

Implementierung einer Geschäftsanwendung auf Basis einer Low-Code Entwicklungsplattform

Masterarbeit

Vorgelegt an der Technischen Hochschule Köln und der Fachhochschule Dortmund
im Studiengang Wirtschaftsinformatik Verbund.

Ausgearbeitet von: Philipp Werner Steinmetzger

Matrikelnummer: 1111 8652

Erster Prüfer: Prof. Dr. Erich Ehse

Zweiter Prüfer: Prof. Dr. Heide Faeskorn-Woyke

Gummersbach im Juli 2019

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis	6
Tabellenverzeichnis	8
Abkürzungsverzeichnis	9
Einleitung	10
1.1 Problemstellung	10
1.2 Forschungsfrage	13
1.3 Aufbau der Arbeit	13
2 Low-Code Programmierparadigma	14
2.1 Historische Einordnung	14
2.2 Programmiersprachen	17
2.3 Modellgetriebenen Softwareentwicklung	18
Unified Modeling Language	19
Business Process Model and Notation	20
Domänenspezifische Sprachen	20
2.4 Computer Aided Software Engineering.....	21
2.5 Rappid Application Development.....	22
2.6 Wiederverwendung und Standardisierung.....	22
3 Low-Code Entwicklungsplattformen.....	23
3.1 Softwareplattform.....	23
3.2 Bereitstellung in der Cloud.....	24
3.3 Abgrenzung No-Code Anwendungsplattformen	26
3.4 Anwendungszielgruppe.....	26
3.5 Aufbau und Komponenten	27
Datenmodellierungswerkzeug.....	28
Oberflächendesigner.....	29
Editoren für Geschäftslogik.....	30
Verwaltung von Bereitstellung	31
3.6 Marktüberblick.....	31
Universelle Lösungen	31
Prozessorientierte Anwendungsfälle.....	32
Datenzentrierte Anwendungsfälle	32
Servicemanagement Anwendungsfälle.....	32
Mobil-native Applikationen	33
4 Anforderungen an Low-Code Entwicklungsplattform.....	34
4.1 Präsentationsschicht.....	34
Zeitgemäße Bereitstellungsart	34
Adaptive Benutzeroberfläche.....	34
Positive Endnutzenerfahrung.....	35
Präsentationslogik.....	35
4.2 Datenschicht	35

Unterschiedliche Informationsarten	35
Datenpersistenz	35
Datenmodellierung.....	36
Datenaustausch und Integration	36
4.3 Geschäftslogik	36
Kalkulation und Aggregation	36
Lesen und Modifikation von Daten.....	37
Datenvalidierungen und Plausibilisierung	37
Geschäftsprozesse	37
Integration externer Geschäftslogik	37
4.4 Querschnitts- und Unterstützungsfunktionen.....	37
Anwendungs- und Datensicherheit.....	38
Benutzerauthentifizierung und Autorisierung	38
Kollaborative Anwendungsentwicklung.....	38
Änderungsnachverfolgbarkeit	38
Anwendungsdebugging	39
Anwendungstest	39
Anwendungsbereitstellung.....	39
Umgebungstrennung	39
Ereignisprotokollierung	40
4.5 Anforderungskatalog.....	40
5 Prototypische Realisierung	41
5.1 Anwendungsfall.....	41
Anforderungen an prototypische Anwendung.....	41
Realisierung mit Low-Code Entwicklungsplattform.....	42
5.2 Oracle Application Express.....	43
Historie und Ausblick	44
Anwendungsentwicklung	46
Lizenzierung.....	46
Systemarchitektur	47
5.3 Installation Low-Code Entwicklungsplattform	48
Installation Oracle Datenbankmanagementsystem	49
Webserver und Dienstschicht Oracle REST Data Services.....	50
Anwendungsmodule Oracle Application Express	50
5.4 Aufruf und Einrichtung Oracle APEX	51
5.5 Datenmodellierung.....	53
5.6 Anlage der Anwendung.....	56
5.7 Seitengestaltung	59
Seitentheme.....	60
Seitengestaltung	61
5.8 Anwendungslogik.....	62

Interaktion innerhalb Anwendung	62
Validierung von Benutzereingaben	63
Anmeldestatus von Schülern	65
Änderungsdokumentation	66
5.9 Beschränkung des Zugriffs	67
5.10 Bereitstellung	70
6 Auswertung	71
6.1 Auswertungsmethodik.....	71
6.2 Anwendungsarchitektur	71
6.3 Präsentationsschicht.....	72
Bewertung der Anforderungen.....	74
6.4 Datenschicht	74
Alternative Werkzeuge zur Datenmodellierung.....	75
Bewertung der Anforderungen.....	77
6.5 Geschäftslogik	77
Interaktion innerhalb Anwendung	77
Eingabevalidierung	77
Geschäftsprozesse	79
Bewertung der Anforderungen.....	80
6.6 Querschnitts- und Unterstützungsfunktionen.....	80
Benutzerauthentifizierung und Autorisierung	81
Anwendungs- und Datensicherheit.....	81
Zusammenarbeit und Kollaboration	83
Anwendungsdebugging	83
Anforderungs- und Projektmanagement	84
Anwendungstest	84
Änderungsnachverfolgbarkeit	86
Umgebungstrennung	86
Anwendungsbereitstellung.....	87
Bewertung der Anforderungen.....	88
6.7 Anwendung des Low-Code Paradigmas.....	89
7 Fazit.....	92
8 Literaturverzeichnis.....	93
9 Anhang.....	96
9.1 Relationales Datenmodell der prototypischen Anwendung	96
9.2 Bildschirmkopien der prototypischen Anwendung	97

Benutzerauthentifizierung	97
Startseite.....	97
Schüler Ansicht	98
Schüler Bearbeitung	98
Schüler Kontakt Bearbeitung	99
Schüler Gruppenzuordnung Bearbeitung	99
Einrichtungen Ansicht	99
Einrichtungen Erfassung mit Validierungsfehler	100
Betreuungsgruppen Ansicht.....	100
Erklärung.....	101

Abbildungsverzeichnis

Abbildung 1: Entwicklung offener Stellen für IT-Fachkräfte seit 2009	11
Abbildung 2: Welche Herausforderungen hat Ihre Organisation bei der Erstellung individueller Unternehmensanwendungen mittels traditioneller Entwicklungsmethoden?	12
Abbildung 3: Haben Low-Code Development Plattformen diese Herausforderungen gelöst?	12
Abbildung 4: UML Klassendiagramm	19
Abbildung 5: Geschäftsprozessdiagramm in BPMN 2.0 Notation	20
Abbildung 6: Komponenten und Architektur einer Low-Code Entwicklungsplattform	27
Abbildung 7: Visuelle Datenmodellierung mit OutSystems Low-Code Entwicklungsplattform	28
Abbildung 8: Visueller Oberflächendesigner OutSystems Low-Code Entwicklungsplattform	29
Abbildung 9: Visuelle Ablaufprogrammierung mit Blockly	30
Abbildung 10: Webbasierte Verwaltung in OutSystems	31
Abbildung 11: Oracle APEX Entwicklertypen	44
Abbildung 12: Komponenten der Low-Code Entwicklungsplattform Oracle APEX	48
Abbildung 13: Einstiegsseite Oracle APEX IDE	52
Abbildung 14: Konzeptionelles Datenmodell nach Chen-Notation	54
Abbildung 15: Oracle APEX Assistent zur Tabellenanlage	55
Abbildung 16: Oracle APEX Objektbrowser	56
Abbildung 17: Seitenstruktur der prototypischen Anwendung	56
Abbildung 18: Oracle APEX Page Designer	57
Abbildung 19: Seite Schüler	58
Abbildung 20: Seite Schüler mit geöffneten Dialogfenster	58
Abbildung 21: Seite Einrichtungen mit interaktivem Grid für Neuanlage und Bearbeitung	59
Abbildung 22: Anpassung Seitentheme mit Oracle APEX Theme Roller	61
Abbildung 23: Gegenüberstellung Formular in Oracle APEX Page Designer und Ausgabe	62
Abbildung 24: Automatische Verzweigung nach Anlage eines Datensatzes	63

Abbildung 25: Gruppenzuordnung Schüler/Gruppe in relationalem Datenmodell.	64
Abbildung 26: Anlage von Werteliste SCHUELER_STATUS in Oracle APEX.....	65
Abbildung 27: Benutzer und Gruppen verwalten in Oracle APEX.....	67
Abbildung 28: Anlage eines Autorisierungsschema mit Oracle APEX IDE	69
Abbildung 29: Einschränkung der Bearbeitung auf Autorisierungsschema	69
Abbildung 30: Standard Seitentemplates Oracle APEX.....	73
Abbildung 31: Testausführung mit Oracle SQL Developer	85

Tabellenverzeichnis

Tabelle 1: Arten von Anwendungsentwicklung.....	16
Tabelle 2: Anforderungskatalog für Low-Code Entwicklungsplattformen.....	40
Tabelle 3: Versionshistorie Oracle APEX.....	45
Tabelle 4: Datenentitäten der prototypischen Implementierung.....	54
Tabelle 5: Datenmapping Dialog Schüler Bearbeiten	61
Tabelle 6: Erweiterung Tabelle SCHUELER für Änderungsdokumentation.....	66
Tabelle 7: Benutzer und Gruppen der prototypischen Anwendung.....	68
Tabelle 8: Autorisierungsschema der prototypischen Anwendung	68
Tabelle 9: Übersicht Anwendungspfad und Bestandteile.....	70
Tabelle 10: Beschreibung der Bewertungskategorien des Anforderungskatalogs	71
Tabelle 11: Bewertung Anforderungen Präsentationsschicht	74
Tabelle 12: Vergleich Quick SQL und Oracle SQL	76
Tabelle 13: Bewertung Anforderungen Datenschicht.....	77
Tabelle 14: Bewertung Anforderungen Geschäftslogik.....	80
Tabelle 15: Bewertung Anforderungen Querschnitts-/Unterstützungsfunktionen .	89
Tabelle 16: Individueller Quelltext in prototypischer Anwendung.....	90

Abkürzungsverzeichnis

APEX	Application Express
BPMN	Business Process Model and Notation
CASE	Computer Aided Software Engineering
CRUD	Create, Read, Update & Delete
DBMS	Databasemanagementsystem
DDL	Data Definition Language
DML	Data Manipulation Language
ERM	Entity Relationship Model
GL	Generation Language
HTML	Hypertext Markup Language
JDBC	Java Database Connectivity
LCDP	Low-Code Development Platform
MDSD	Model Driven Software Development
UI	User Interface
OMG	Object Management Group
ORDS	Oracle REST Data Services
PaaS	Platform-as-a-Service
RAD	Rapid Application Developement
REST	Representational State Transfer
RPM	RedHat Package Manager
SaaS	Software-as-a-Service
SSH	Secure Shell
SQL	Structured Query Language
UML	Unified Modeling Language
WYSIWYG	What You See Is What You Get

Einleitung

Der Umgang und die Erwartungshaltung an Informationstechnologie hat sich in den letzten Jahren erheblich verändert. Statt den Weg in ein Kaufhaus zu nehmen, wird mit wenigen Klicks auf Online-Shopping Portalen bestellt. Die nächste Reise wird nicht im Reisebüro ausgewählt, sondern online über Vergleichsportale verglichen und gebucht. Die Pizza wird nicht in der Pizzeria abgeholt, sondern per Mobiltelefon App bei einem Lieferdienst bestellt. Man kauft kein Auto mehr, sondern mietet Mobilität bei Carsharing Dienstleistern. Und alles wird in kürzester Zeit, meist frei Haus, geliefert.

Die Digitalisierung ermöglicht deutliche Wettbewerbsvorteile. Gleichzeitig wächst durch sie aber auch der Druck auf Unternehmen: Würden Kunden lange auf ein Angebot warten, wenn sie dieses bei der Konkurrenz online selbst abrufen können? Es gibt kaum eine Branche, die nicht auf die ein oder andere Art von der Digitalisierung beeinflusst wird. Dadurch steigt auch der Bedarf an individuellen Geschäftsanwendungen. Unternehmen in allen Branchen wollen damit ihre Arbeitsabläufe vereinfachen, bessere Services anbieten und so Ihre Wirtschaftlichkeit erhöhen.

Jedoch führt dieser Wettlauf um digitalisierte Geschäftsprozesse auch dazu, dass herkömmliche Entwicklungsansätze mit den zunehmenden Anforderungen nicht mehr mithalten können. Die IT- und Entwicklungsabteilungen müssen immer schneller benutzerdefinierte Anwendungen entwickeln und bereitstellen. Den Marktanalysen von Gartner und Forrester zufolge schlägt genau aus diesem Grund nun die Stunde der Low-Code Entwicklungsplattformen.¹

1.1 Problemstellung

Für Unternehmen wird es zunehmend schwieriger, geeignete Fachkräfte für die IT zu finden. Ende 2018 zählte der Branchenverband Bitkom 82.000 offene Stellen für IT-Spezialisten. Das entspricht einem deutlichen Zuwachs von 49 Prozent im Vergleich zum Vorjahr. In den Unternehmen sind Software-Entwicklern, Anwendungsbetreuer und Administratoren am stärksten gefragt, gefolgt von IT-Sicherheitsexperten. Die IT-Abteilungen in Anwenderunternehmen sind aufgrund des Fachkräftemangels schon jetzt häufig komplett ausgelastet.²

¹ Vgl. Rymer & Koplowitz, Low-Code Development Platforms For AD&D Professionals

² Vgl. Rohleder, Der Arbeitsmarkt für IT-Fachkräfte

Anzahl zu besetzender IT-Stellen in Gesamtwirtschaft

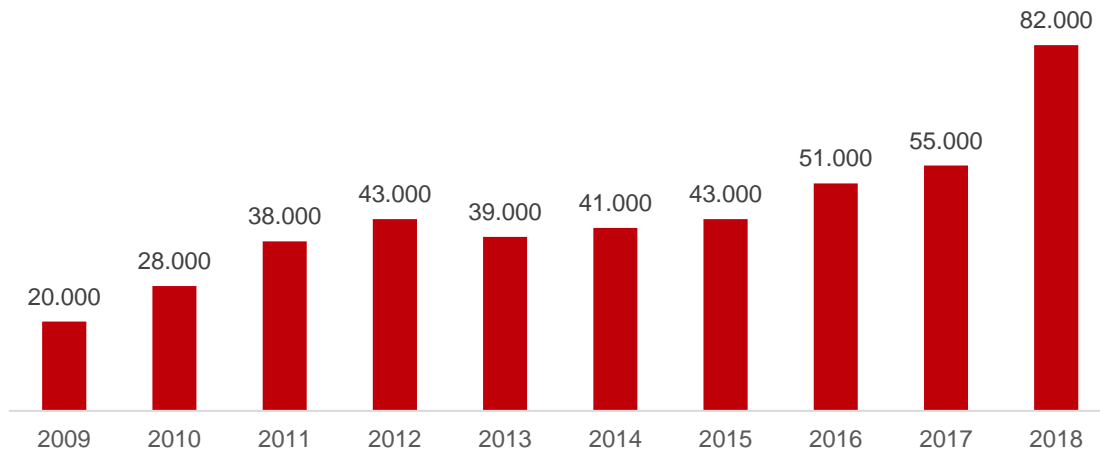


Abbildung 1: Entwicklung offener Stellen für IT-Fachkräfte seit 2009 (Quelle: Rohleder, 2018)

Dazu kommt, dass die Ansprüche der Anwender heute so hoch sind wie noch nie. Aus dem privaten Gebrauch sind Mitarbeiter und Kunden Anwendungen gewohnt, die mobil, einfach und schnell funktionieren. Lange Ladezeiten führen ebenso zu Frust wie lange Wartezeiten bei Bestellungen. Der vom Journalisten und Blogger Peter Glaser geprägte Begriff der „Sofortness“ bezeichnet die Erwartungshaltung der Gesellschaft, die „sofort“ als den einzig akzeptablen Zeitrahmen darstellt. Von Sascha Lobo wurde dies auch als „technologische Seite der Ungeduld“ und die „Digitale Ungeduld“ beschrieben.³

Diese Erwartungshaltung gilt in besonderem Maße für die Arbeitnehmer der Millennial-Generation, also der Bevölkerungsgeneration, die im Zeitraum der frühen 1980er bis zu den späten 1990er Jahren geboren wurden. Im Privatleben ist Amazon das Kaufhaus, Spotify der Plattenladen und Netflix die Videothek. Durch die ständige Interaktion mit digitalen Services im Privaten entsteht eine hohe Erwartungshaltung an betriebliche Prozesse hinsichtlich Geschwindigkeit und Unmittelbarkeit.

Diese Faktoren führen dazu, dass die zeitnahe Entwicklung von Unternehmensanwendungen als größte Herausforderung für IT-Teams wahrgenommen wird. In aktuellen Veröffentlichungen wird der Zeitdruck als noch größeres Problem als die mangelnde Flexibilität beschrieben.⁴

³ Vgl. Lobo, Digitale Ungeduld

⁴ Vgl. Rymer & Koplowitz, Low-Code Development Platforms For AD&D Professionals

Herausforderung bei Erstellung von Geschäftsanwendungen

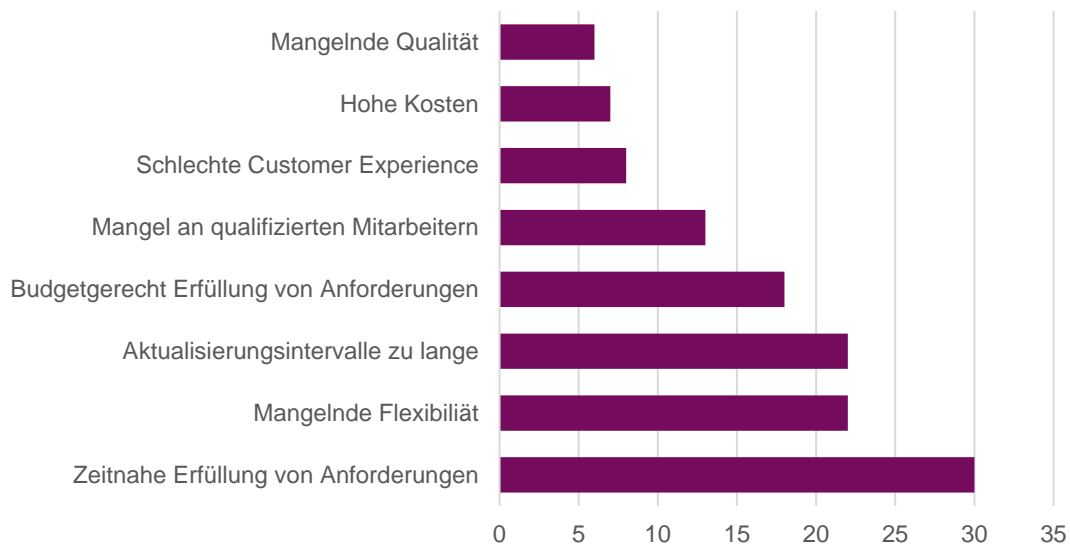


Abbildung 2: Welche Herausforderungen hat Ihre Organisation bei der Erstellung individueller Unternehmensanwendungen mittels traditioneller Entwicklungsmethoden? (Quelle: Rymer & Koplowitz, 2019)

Als potenzielle Lösung dieser Herausforderungen werden Low-Code Entwicklungsplattformen gehandelt. Diese Plattformen versprechen die wesentlich schnellere und damit effizientere Entwicklung von Geschäftsanwendungen. Und laut den für die Forrester Wave befragten Leitern von IT- und Entwicklungsabteilungen lösen sie dieses Versprechen auch ein. Über 50 Prozent der Befragten geben an, der Einsatz einer Low-Code Plattform habe Ihre Situation „signifikant verbessert“. Im Folgenden erfahren Sie, was eine Low-Code Plattform ausmacht und welche Vorteile sie im Vergleich zum klassischen Entwicklungsansatz bietet.⁵

Haben Low-Code Plattformen die Probleme gelöst?

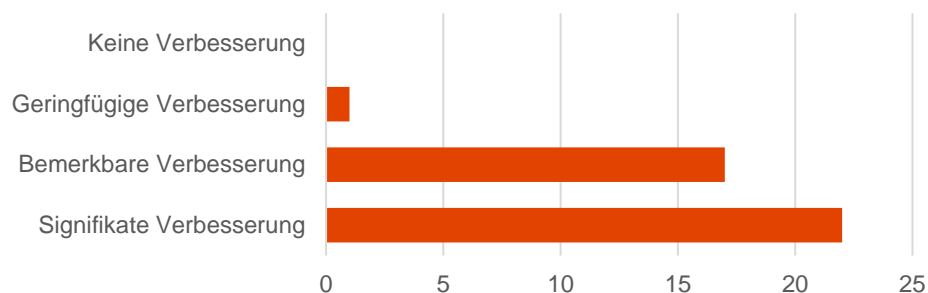


Abbildung 3: Haben Low-Code Development Plattformen diese Herausforderungen gelöst? (Quelle: Rymer & Koplowitz, 2019)

⁵ Rymer & Koplowitz, Low-Code Development Platforms For AD&D Professionals

1.2 Forschungsfrage

Diese Arbeit setzt sich mit dem Low-Code Entwicklungsparadigma auseinander. Dabei soll die Frage beantwortet werden, ob aktuell auf dem Markt befindliche Low-Code Entwicklungsplattformen dazu geeignet sind, gängige Anforderungen an Geschäftsanwendungen umzusetzen.

Durch die Beantwortung der Forschungsfragen soll diese Masterarbeit weiter dazu dienen, einen Überblick über Funktionalitäten von Low-Code Entwicklungsplattformen zu gewinnen und deren praktische Nutzbarkeit anhand einer prototypischen Realisierung zu validieren.

1.3 Aufbau der Arbeit

Zunächst soll das Low-Code Programmierparadigma definiert und im Kontext zu Methoden wie der modellgetriebenen Softwareentwicklung und Rapid Application Development dargestellt werden.

Im darauffolgenden Abschnitt liegt der Fokus auf Plattformen die eine ganzheitliche Low-Code Entwicklung als Lösung offerieren. Ein Überblick über Funktionalität und Angebot in diesem Bereich soll geschaffen und die Low-Code Entwicklungsplattformen nach ihrer Funktionalität eingeordnet werden.

Anhand gängiger Entwicklungsansätze unter Nutzung eines Schichtenmodells zur Realisierung von Geschäftsanwendungen, sollen Anforderungen an die Low-Code Entwicklungsplattformen skizziert und in Form eines Anforderungskatalogs gesammelt werden.

Unter Nutzung der Low-Code Entwicklungsplattform Oracle Application Express erfolgt die Realisierung eines Prototyps einer Geschäftsanwendung. Die fachlichen Anforderungen für diese Geschäftsanwendung basieren auf ein Praxisprojekt.

Auf Basis der Erkenntnisse aus der prototypischen Realisierung soll geprüft werden, ob die spezifischen Anforderungen sowie die allgemeinen Anforderungen gemäß des Anforderungskatalogs zufriedenstellend mit der Low-Code Entwicklungsplattform gelöst werden konnten. Weiter soll analysiert werden, ob die Quelltextmenge tatsächlich gering ausgefallen ist und die Einordnung des Produkts Oracle Application Express als Low-Code Entwicklungsplattform gerechtfertigt ist.

Im abschließenden Kapitel erfolgt die Prüfung ob im Rahmen dieser Ausarbeitung die Forschungsfragen beantwortet werden konnte.

2 Low-Code Programmierparadigma

Das Low-Code Programmierparadigma zeichnet sich dadurch aus, dass die Menge an erforderlichen Quelltext zum Erlangen der gewünschten Anwendung gering ausfällt. Durch die geringe Menge an erforderlichen Quelltext für eine Anforderung soll der Gesamtaufwand für die Entwicklung einer Geschäftsanwendung einschließlich aller Entwicklungsphasen, wie Entwurf, Implementierung und Test, verringert werden.⁶

Es ist davon auszugehen, dass unterschiedlichste Lösungsansätze zu dem heutigen Low-Code Programmierparadigma führten. Neben der Entwicklung von neuen Generationen von Programmiersprachen, sind insbesondere die ähnlichen Ansätze des Rapid Application Developments (RAD), die modellgetriebene Softwareentwicklung und Computer Aided Software Engineering (CASE) als Treiber hierfür aufzuführen.⁷

2.1 Historische Einordnung

Die Effizienz der Softwareentwicklung rückte mit der Softwarekrise in den 1960er Jahren initial in den Fokus des IT-Managements. Erstmals überstiegen in der Softwarekrise die Kosten für Entwicklung von Software die Kosten der Hardware. Zugleich konnten die Anforderungen, die an Anwendungsprogramme gestellt wurden, mit den zu diesem Zeitpunkt gängigen Methoden und Werkzeugen der Softwareentwicklung nicht oder nicht ausreichend effizient gelöst werden. Es folgte das Scheitern erster großer Softwareprojekte. Nach Dijkstra lag die Ursache hierfür darin begründet, dass die Komplexität der Softwareentwicklung in Relation zu den Möglichkeiten der Computer steht und diese Komplexität nicht mehr beherrschbar sei.⁸

In Folge vergleicht Martin die Situation der Computerindustrie mit den Anfängen der Telefonie. Wenngleich das Potential und die Vorteile der Telefonie schon früh deutlich waren, war die Telefonie zunächst nur einer sehr begrenzten Anzahl von Teilnehmern, ausschließlich Unternehmen lokal begrenzt, zugänglich. Einschränkender Faktor war hierbei, dass die Vermittlung von Telefonleitungen anfangs manuell durch Personal in den Vermittlungsstellen durchgeführt wurde. Mit dem Anstieg der Teilneh-

⁶ Vgl. Rentrop & Augsten, Low-Code-Entwicklung

⁷ Vgl. Jahn, Low-Code Development Plattform

⁸ Vgl. Dijkstra, The Humble Programmer

mer war ein proportionaler Anstieg des erforderlichen Personals in den Vermittlungsstellen verbunden. Die damalige Vision, einer weiten Verbreitung der Telefonie in sämtlichen Büros war durch diese Abhängigkeit und des personellen Mehrbedarfs nur schwer in die Tat umzusetzen. Erst durch die Erfindung von automatischen Vermittlungsstellen und Wählsystemen von Strowger konnte die Telefonie einer breiten Anzahl von Nutzern zugänglich gemacht werden und damit ihr volles Potential entfalten.⁹

Nach Martin können hier Parallelen zur Computerindustrie gezogen werden. Während bei der Telefonie das manuelle Vermitteln von Gesprächen der einschränkende Faktor war, besteht dieser in der Computerindustrie durch hohen Personalbedarf aufgrund ineffizienter Softwareentwicklung. Erst durch effizientere Softwareentwicklung kann laut Martin die Technologie ihr volles Potential ausschöpfen. Martin stellte im Jahr 1982 Berechnungen auf, dass unter der Annahme einer kontinuierlichen Wachstumsrate der Computerindustrie von 25 Prozent je Jahr, bei gleichbleibender Effizienz der Softwareentwicklung bereits im Jahr 1990 ein Bedarf von rund 28 Millionen Entwicklern erforderlich gewesen wäre.¹⁰ Aus damaliger Sicht war es erforderlich, dass die Produktivität der Anwendungsentwicklung um mindestens die zweifache Größenordnung ansteigt.

Als möglicher Ausweg aus der Softwarekrise galten damals strukturierte Programmiersprachen, die die Entwicklung von Anwendungen vereinfachen und deren Konzepte sich in heute gängigen Programmiersprachen wiederfinden. Auch damals entwickelte Methoden wie „Literal Programming“, das Schreiben von Computerprogrammen in für Menschen leserlicher Sprache, galt vielversprechend und ist in heute gängigen Ansätzen wie der modellgetriebenen Softwareentwicklung wiederzufinden.

Martin skizzierte drei unterschiedliche Arten von Anwendungsentwicklung:

- Konventionelle Anwendungsentwicklung, die unter den beschriebenen Problemen und Herausforderungen leidet.
- Einsatz von Anwendungsgeneratoren für die Erstellung von Anwendungsprototypen. Der Endnutzer interagiert hierbei mit den Prototypen und ein Analyst modifiziert es solange bis es geeignet für die Anwendungsentwicklung ist. Das

⁹ Vgl. Martin, Application development without programmers, S. 2

¹⁰ Vgl. Martin, Application development without programmers, S. 1–4

Prototypingwerkzeug verringern den Bedarf einer separaten Anforderungsdokumentation soweit wie möglich. Aus dem Werkzeug können ggf. Quelltext Bausteine für die tatsächliche Anwendung generiert werden.

- Nutzung eines Anwendungsgenerators oder ähnlicher Software, um die gesamte Anwendung zu entwickeln. Der Prototyp wird zur eigentlichen Anwendung. Es bedarf keiner separaten Programmierung.

Diese unterschiedlichen Arten von Anwendungsentwicklung sind in heute gängigen Paradigmen der Low-Code Softwareentwicklung wiederzufinden und stellen wie in Tabelle 1 dargestellt unterschiedliche Vor- und Nachteile in den Phasen der Anwendungsentwicklung dar.

	Konventionelle Anwendungsentwicklung	Nutzung von Prototyping Werkzeug	Anwendungsentwicklung ohne Programmierung
Anforderungsanalyse	Zeitaufwändig, oft stark formal. Meist zeitlich versetzt zu Entwicklung	Stimulierung der Vorstellungskraft des Endnutzers durch visuellen Prototyp, gemeinsame Erarbeitung ggf. mit Business Analyse	Stimulierung der Vorstellungskraft des Endnutzers, Entwicklung von eigenen Anforderungen durch Endnutzer.
Fachkonzept	Ausführliches Dokument, ggf. fern von technischer Realität	Prototyp als Anforderung, präzise und getestet.	- entfällt -
Freigabe durch Auftraggeber	Möglicherweise nicht im Detail für Endnutzer klar. Feinheiten unklar.	Nutzer sieht Ergebnis vor Freigabe und kann dies nach Bedarf anpassen.	Kein formaler Freigabeprozess, da Entwicklung und Anpassung durch Anwender selbst möglich
Entwicklung & Test	Geringe Geschwindigkeit, hohe Kosten. Bei mehr Nachfrage als Ressourcen langsam.	Aus Prototyp wird Anwendung entwickelt, ggf. Nutzung generierten Quelltextes möglich. Schneller und reduziert Fehlerfälle.	Umsetzung ist schnell möglich. Test kann ggf. reduziert werden.
Dokumentation	Erforderlich und zeitaufwändig.	Kann ggf. aus Prototypen teilweise generiert werden	Weitestgehend automatisiert, Anwendung ist die Dokumentation.
Anpassungen	Erfordert neue Beauftragung.	Erfordert neue Beauftragung.	Kontinuierlicher Prozess, Anpassungen durch Endnutzer nach Bedarf möglich.

Tabelle 1: Arten von Anwendungsentwicklung (Quelle: Martin)

Nach Meinungen von Martin ist die Softwarekrise bis heute nicht beendet, sondern existiert in anderer Qualität und anderem Ausmaß fort.¹¹

¹¹ Vgl. Martin, Application development without programmers, S. 3

2.2 Programmiersprachen

Der naheliegendste Ansatz zur Verringerung des erforderlichen Quelltextes liegt im Aufbau und der Struktur der verwendeten Programmiersprache selbst. Je einfacher und kürzer der Entwickler an das von ihm beabsichtigte fachliche Ziel gelangt, umso weniger Quelltext ist erforderlich. Die Entwickler von Programmiersprachen bemühen sich jeher, die Programmiersprachen so weiterzuentwickeln, dass mit möglichst wenig Eingaben das gewünschte Ziel erreicht werden kann.

In Analogie zu Hardwaregenerationen werden Programmiersprachen in Sprachgenerationen (englisch Generation Language, kurz GL) unterteilt. Bei Betrachtung dieser Sprachgenerationen wird deutlich, wie durch die Weiterentwicklung der Programmiersprachen der erforderliche Quelltext bereits verringert wurde.

Die erste Generation der Programmiersprache bezeichnet die Maschinensprache. Sie ist prinzipiell binär, kann jedoch auch hexadezimal codiert sein. Durch ihren unverständlichen und unübersichtlichen Aufbau können komplexere Programmierungen kaum realisiert werden. Bei der Assemblersprache (2GL) wird zu den binären Codes so genannte „Mnemonics“ Kürzel zugeordnet. Der Quelltext, auch Assemblercode genannt, wird durch den sogenannten Assembler direkt in ausführbare Maschinensprache (1GL) umgewandelt. Auch in den nachfolgenden Generationen ging es immer darum, eine Sprache in die nächst Niederer zu übersetzen.

Die gängigste Generation ist auch heute noch 3GL: die prozedurale Programmierung. Bekannte Sprachen sind beispielsweise C++, Java oder JavaScript. Da 3GL zu den höheren Programmiersprachen gehört, muss diese vor der Ausführung durch einen Compiler in einen Maschinencode umgewandelt werden.

Während 3GL vor allem die Einführung von standardisierten Kontrollstrukturen geprägt hat, haben Programmierumgebungen der vierten Generation (4GL) das Ziel, möglichst schnell und mit möglichst wenig Codezeilen für einen bestimmten Anwendungsfall Funktionen oder aber auch komplette Anwendungen bereitstellen zu können. Dieser Trend setzte in den 1980er Jahren ein und kann somit als erster Impulsgeber für die heutige Low-Code Bewegung gesehen werden.

Der Begriff 4GL ist nicht exakt definiert und wird vor allem für Marketingzwecke eingesetzt. Die als 4GL eingeordneten Programmiersprachen sind oft auf bestimmte Thematik festgelegte, anwendungsspezifische Skript- oder Abfragesprachen wie z.B. die Datenbanksprache Strcutured-Query-Language (SQL). Einige fortgeschrittene Sprachen der dritten Generation wie Python, Ruby und Perl kombinieren 4GL Fähigkeiten in einer grundsätzlichen 3GL Sprachumgebung. Darüber hinaus wurden teils

Programmbibliotheken entwickelt, die Fähigkeiten von 3GL Sprachen als 4GL Sprache kapseln. Dies führte zu einer Verschmelzung von 3GL und 4GL Sprachen. Die Bezeichnung und Definition 4GL wurde von James Martin und Richard Murch erstmals 1982 erwähnt.¹²

Während die Programmiersprachen der vierten Generation dazu dienen sollen, spezifische Anwendungen so effizient wie möglich zu entwickeln, sollen Sprachen der fünften Generation so aufgebaut sein, dass der Computer die Probleme der Anwendungsfälle ohne Programmierung löst.¹³ Sprachen der fünften Generation werden bisher überwiegend im Feld der künstlichen Intelligenz, z.B. OPS5 und Mercury, eingesetzt und sind bisher in der Praxis bei Entwicklung von Geschäftsanwendungen noch eher wenig relevant.

2.3 Modellgetriebenen Softwareentwicklung

Als weiterer Ansatz die Entwicklung von Anwendungen so einfach und effizient wie möglich zu machen, kann die modellgetriebene Softwareentwicklung (Model Driven Software Development, kurz MDSD) angesehen werden. Während auch noch in den 4GL Programmiersprachen Quelltext zur Definition angewandt wird, setzt MDSD auf formale Fachmodelle, aus denen automatisiert lauffähige Software erzeugt wird.¹⁴

Die Software beruht hier also weniger auf der Programmiersprache, sondern vielmehr auf Modellsprachen. Diese Modellsprachen haben den Vorteil, dass sie fachspezifisch sind und somit auch von Fachbereichen leichter beherrschbar sind. Als Kernidee der modellgetriebenen Softwareentwicklung kann die Absicht verstanden werden, den Abstand zwischen Fachbereich und Entwicklung mit der Modellierungssprachen zu verkleinern. Modelle sollen einen ganzheitlichen und gemeinsamen Blick auf die Domäne ermöglichen, die sowohl den technischen als insbesondere auch den fachlichen Anforderungen Rechnung trägt.

Generatoren und Interpreter sowohl zur Entwicklungs- als auch zur Laufzeit werden anschließend verwendet, um den eigentlichen Anwendungsquelltext soweit wie möglich aus dem Modell zu generieren beziehungsweise zu interpretieren. Nur der Anteil, der nicht als Modell darstellbar ist, wird zusätzlich manuell und oft textuell kodiert.

¹² Vgl. Martin, Application development without programmers, S. 4–6

¹³ Vgl. Grigonis, Generation Computers

¹⁴ Vgl. Whittle, Hutchinson & Rouncefield, The State of Practice in Model-Driven Engineering

In der Anwendungspraxis, aber auch in den betrachteten Low-Code Entwicklungsplattformen, sind überwiegend zwei Modellierungssprachen, die Unified Modeling Language und die Business Process Model Notation aus der modellgetriebenen Softwareentwicklung vorzufinden.

Unified Modeling Language

Die Unified Modeling Language (UML) definiert eine allgemein verwendbare Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und Systemen. Sie wird von der Object Management Group (OMG) entwickelt und ist sowohl von ihr als auch von der International Organization for Standardization (ISO) genormt.¹⁵ Im Sinne einer Sprache definiert UML dabei Bezeichner für die meisten bei einer Modellierung wichtigen Begriffe und legt mögliche Beziehungen zwischen diesen Begriffen fest. UML definiert grafische Notationen für diese Begriffe und für Modelle statischer Strukturen und dynamischer Abläufe, die man mit diesen Begriffen formulieren kann.¹⁶

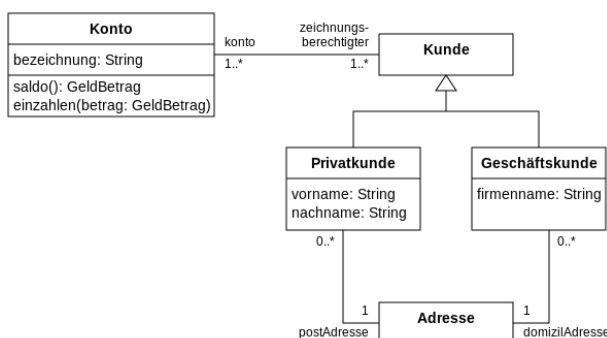


Abbildung 4: UML Klassendiagramm (Quelle: Kecher, Salvanos & Hoffmann-Elbern)

Der erste Kontakt zu UML besteht häufig darin, dass Diagramme in UML im Rahmen von Softwareprojekten zu erstellen, zu verstehen oder zu beurteilen sind: Projektauftraggeber und Fachvertreter prüfen und bestätigen zum Beispiel Anforderungen an ein System, die Business Analysten in Anwendungsfalldiagrammen in UML festgehalten haben; Softwareentwickler realisieren Arbeitsabläufe, die Business Analysten in Zusammenarbeit mit Fachvertretern in Aktivitätsdiagrammen beschrieben haben; Systemingenieure implementieren, installieren und betreiben Softwaresysteme basierend auf einem Implementationsplan, der als Verteilungsdiagramm vorliegt.

¹⁵ Vgl. International Organization for Standardization, Object Management Group Unified Modeling Language

¹⁶ Vgl. Kecher, Salvanos & Hoffmann-Elbern, UML 2.5, S. 20–21

Business Process Model and Notation

Die Business Process Model and Notation (BPMN) ist eine grafische Spezifikations-
sprache zur Modellierung und Dokumentation von Geschäftsprozessen und Arbeits-
abläufen. Der Standard definiert neben der Notation der einzelnen Symbole auch de-
ren Semantik. In Abbildung 5 wird exemplarisch ein BPMN 2.0 Diagramm dargestellt,
dass den Prozess einer Pizza-Bestellung einschließlich Lieferung abbildet.¹⁷

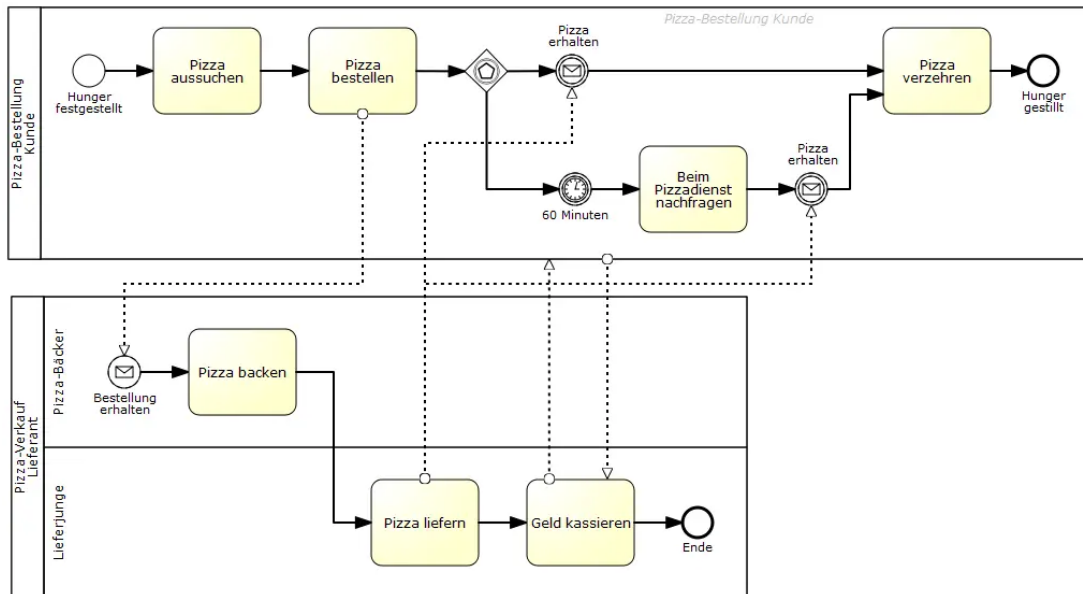


Abbildung 5: Geschäftsprozessdiagramm in BPMN 2.0 Notation (Quelle: Neumann)

Durch die Eindeutigkeit der verwendeten Symbole und deren Semantik existiert ne-
ben der grafischen Notation der XML basierter Standard BPMN 2.0 XML. Auf Basis
dieses XML-Standards können Geschäftsprozesse in Anwendungen einfach imple-
mentiert werden, ohne diese individuell programmieren zu müssen. Programmiblio-
theken wie z.B. das quelloffene Softwareprojekt Flowable für die Programmierspra-
che Java, ermöglichen die Ausführung der XML basierten Prozessbeschreibungen.
Die sonst erforderliche Programmierung der Prozesse in der Programmiersprache
entfällt somit.¹⁸

Domänenspezifische Sprachen

Neben den vorgestellten standardisierten und weit verbreiteten Modellierungsspra-
chen UML und BPMN existieren jedoch auch domänenspezifische Sprachen (engl.

¹⁷ Vgl. Neumann, BPMN 2.0 für eine bessere Zusammenarbeit zwischen Fachabteilung und IT

¹⁸ Vgl. Chandrakant, Introduction to Flowable

domain specific languages, DSL). Auch diese domänenspezifischen Sprachen ermöglichen eine einfache Beschreibung von Funktionalitäten durch Fachbereiche und eine anschließende Generierung des eigentlichen Anwendungs Quelltextes.

Da diese domänenspezifischen Sprachen jedoch nicht unbedingt branchen- oder fachgebietsbezogen sein müssen, zeigt das Projekt Open Standard Business Platform (OSBP) der Eclipse Foundation. Bei OSBP handelt es sich um eine quelloffene Softwarefabrik. Die Entwicklung mit der Fabrik erfolgt modellgetrieben durch den Einsatz von 26 domänenspezifische Sprachen, z.B. für Benutzeroberflächen, Berichte und Datenmodelle. OSBP persistiert diese Modelle, sodass Designer und Entwickler jederzeit Veränderungen vornehmen können. Aus diesen Modellen können durch OSBP dann die Anwendungen generiert werden.¹⁹

2.4 Computer Aided Software Engineering

Als weiterer Lösungsansatz für Low-Code Softwareentwicklung kann die rechnergestützte Entwicklung von Software (engl. Computer Aided Software Engineering, CASE) gesehen werden. Wesentlich hierbei ist, dass die Entwicklung von Software durch den Einsatz von speziellen Werkzeugen, s.g. CASE-Tools, hier vereinfacht und beschleunigt werden soll. Differenziert wird hierbei nach Upper Case Tools, deren Fokus auf die Schritte der Analyse und Entwurf, sowie Lower Case Tools, deren Fokus auf Codegenerierung, Testen und Debugging liegt.²⁰

Aktuell können Integrierte Softwareentwicklungsumgebungen wie Eclipse mit entsprechenden Erweiterungen als CASE Werkzeuge verstanden werden. Als weitere, frei verfügbare IDE kann NetBeans genannt werden. Vertreter aus der kommerziellen Branche sind bspw. IBM Rational und Microsoft Visual Studio. CASE-Tools werden mittlerweile ebenfalls für spezifische Anwendungsdomänen, etwa die Entwicklung mobiler Anwendungen (Android Developer Tools von Google oder Xcode von Apple) angeboten.

¹⁹ Vgl. Loetz, OSBP – die erste Software-Fabrik bei Eclipse

²⁰ Vgl. Dewanto & Klein, Low-Code == Low Quality?

2.5 Rappid Application Development

Während CASE gezielt einzelne Entwicklungsschritte unterstützt, geht der Rappid Application Development (RAD) Ansatz weiter. Hierbei sollen Entwickler gesamte Anwendungen zusammenstellen können und hierfür kaum Quelltext erforderlich sein. Bei dieser Art von Anwendungen steht die grafische Oberfläche meist im Mittelpunkt und die fertigen Anwendungen werden als Gesamtanwendung ausgeliefert. Bekannte Vertreter dieses Lösungsansatzes sind die Produkte Oracle Forms, Visual Basic, Delphi und VisualFoxPro für datenbankbasierte Geschäftsanwendungen.²¹

Der RAD Ansatz wird auch in branchenspezifischen Lösungen teils sehr erfolgreich angewandt, beispielsweise im Bereich der Mess-, Regel- und Automatisierungstechnik. Die Software LabVIEW der Firma National Instruments ist erstmals 1986 erschienen und ermöglicht mit ihrem grafischen Programmiersystem Fachanwendern die einfache Entwicklung von virtuellen Messinstrumenten.

2.6 Wiederverwendung und Standardisierung

Ein weiterer Ansatz, der in der Softwarearchitektur schon lange gelebt wird, ist die Wiederverwendung von Programmbestandteilen in Form von Programmpaketen und Frameworks. Während der Anteil von wiederverwendeten Softwarepaketen in den Jahren bis 1980 bei maximal bis zu vier Prozent lag, ist mittlerweile 20 Prozent und mehr des Quelltextes ein vorgefertigtes Programmpaket.²²

Dies beschleunigt die Softwareentwicklung deutlich. Im Umfeld von Webapplikationen sind hier einerseits Frameworks wie z.B. das Spring Framework zu nennen, die serverseitige Komponenten wie z.B. die Datenspeicherung und Geschäftslogik vereinfacht abbilden. Für die Erstellung von Oberflächen haben sich Frameworks wie z.B. das Google Web Toolkit (GWT) von Google bewährt, die es ermöglichen Benutzeroberflächen zu erstellen.

²¹ Vgl. Dewanto & Klein, Low-Code == Low Quality?

²² Vgl. Martin, Application development without programmers, S. 88–89

3 Low-Code Entwicklungsplattformen

Im vorangegangenen Kapitel wurden unterschiedliche Lösungswege und Ansätze aufgezeigt, um mit weniger Quelltext die Entwicklung von Geschäftsanwendungen durchführen zu können. Die meisten dieser Ansätze reichen etliche Jahre zurück und können aus heutiger Sicht nicht mehr als sonderlich innovativ angesehen werden. Dennoch kann aktuell beobachtet werden, wie diese spezielle Form der Anwendungsentwicklung erst jetzt zunehmend an Relevanz für die Entwicklung von Geschäftsanwendungen aufnimmt.

3.1 Softwareplattform

Der jetzige Durchbruch von Low-Code Entwicklungsansätzen und dessen Technologien ist insbesondere auf den Plattformcharakter zurück zu führen. In Form von Low-Code Entwicklungsplattformen werden Systeme bereitgestellt, die die benötigten Funktionalitäten zur Entwicklung von Geschäftsanwendungen nach dem Low-Code Programmierparadigma vorgefertigt bereitstellt. Die initiale Einrichtung und Konfiguration der oft komplexen Low-Code Entwicklungstechnologien z.B. von CASE Werkzeugen oder Umwandlern von Modellen, entfällt somit.²³ Dabei versprechen diese Low-Code Entwicklungsplattformen, intelligent genug gebaut zu sein, dass Funktionalitäten nur noch zusammen geklickt werden müssen, ohne dass sich um technische Belange im Detail gekümmert werden muss. Die Erstellung der Anwendung einschließlich deren Betrieb erfolgt somit auf Basis dieser vorgefertigten und bereitgestellten Softwareentwicklungsplattform.

Als Softwareentwicklungsplattform wird nach Forrester Research hierbei eine Umgebung verstanden, die alle erforderlichen Funktionalitäten bereitstellt, um ein Informationssystem zu entwickeln, auszuführen und zu verwalten.²⁴

Bei Betrachtung der Low-Code Entwicklungsplattformen kann grob nach zwei Arten differenziert werden. Die erste Gruppe von Plattformen orientiert sich stark am Ansatz der modellgetriebenen Softwareentwicklung. Die Plattformen nutzen an zentraler Stelle Modelle, um die Funktionalität der Software zu beschreiben. In diesem Fällen ist kein oder nur sehr wenig Quelltext erforderlich, um die Anwendung zu erstellen. In

²³ Vgl. Hendriks, The selection process of model based platforms, S. 15

²⁴ Vgl. Rymer & Koplowitz, Low-Code Development Platforms For AD&D Professionals

der Softwareplattform wird zentral ein Modell der Anwendung erstellt, einzelne Segmente der Anwendung werden verlinkt und die Anwendung erstellt. Nach Hendriks zielen diese Plattformen vor allem auf „new smart developers“. Dies sind Personen, die nicht unbedingt Programmiererfahrung mitbringen, sondern Personengruppen wie Geschäftsprozessverantwortliche oder IT-Architekten. Die Plattformen generieren den Quelltext basierend auf dem, was der Benutzer modelliert hat. Wenn ein Benutzer beispielsweise ein Eingabefeld mit einem spezifischen Datenbankfeld verknüpft, stellt die Plattform sicher, dass alles Erforderliche unternommen wird, dass die Benutzereingabe in der Anwendung tatsächlich in die Datenbank transferiert wird. Durch diesen Ansatz wird das erforderliche technische Wissen vom Anwender in die Plattform verlagert, der Anwender muss sich nicht mehr mit der technischen Realisierung auskennen, um eine Anwendung dieser Art zu erstellen.²⁵

Als zweite Art von Low-Code Entwicklungsplattformen sind Anwendungen zu sehen, die eher als webbasierte Variante einer Entwicklungsumgebung angesehen werden können. Es handelt sich hierbei um noch sehr technologielastriche Plattformen, die breites Programmierwissen erfordern. Diese Art von webbasierten Entwicklungsumgebungen bieten deutlich mehr individuelle Funktionalitäten, erfordern jedoch mehr Quelltext.

3.2 Bereitstellung in der Cloud

Als weitere Eigenschaft, die die Renaissance der Low-Code Programmierparadigmen sicherlich mitbeflügelt, ist der in den letzten Jahren angestiegene Einsatz von Cloud-Lösungen. Auch die Low-Code Entwicklungsplattformen können als Cloud-Lösungen in Form von sogenannten Platform-as-a-Service (PaaS) Umgebungen genutzt werden. Die Besonderheit hierbei ist, dass der gesamte Betrieb der Entwicklungsplattform und der eigentlichen Anwendung durch einen externen Dienstleister erfolgt.

Nach Richardson und Rymer hat diese Nutzung gewisse Vorteile:²⁶

- **Einfachheit:** Die Nutzung einer PaaS Umgebung ist einfacher als die sonst selbst erforderliche Bereitstellung, Konfiguration und Verwaltung aller erforderlichen Komponenten. Meist hat der Betreiber der PaaS Umgebung auch schon alle relevanten Entscheidungen zur Konfiguration getroffen. Da nicht mit all den

²⁵ Vgl. Hendriks, The selection process of model based platforms, S. 15–16

²⁶ Vgl. Richardson & Rymer, Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms

technischen Hintergründen beschäftigt werden muss, beschleunigt dies die Time-To-Market für eine neue Anwendung.

- **Skalierbarkeit:** Die Skalierung von Anwendungen ist in PaaS Umgebungen meist problemlos möglich. Die meisten Anbieter bieten den Nutzern an, die Geschäftsanwendung auf Infrastruktur des Anbieters auszuführen, die eine Skalierung berücksichtigt. Für den Anbieter bietet dies die Möglichkeit die Rechenkapazität für all seinen Kunden gemeinsam in der Cloud zu nutzen. Durch unterschiedliche Nutzungszeiten und Spitzenlasten kann der Anbieter die Rechenkapazitäten somit effizient auslasten.
- **Kosten:** Bei PaaS Verträgen muss durch den Anwender meist nur in Abhängigkeit von der tatsächlichen Nutzung gezahlt werden. Es entstehen somit keine fixen Lizenzkosten z.B. für eingesetzte Softwarepakete oder Entwicklungssoftware. Der Anwender muss auch keine Server für Entwicklung, Test oder Betrieb selbst betreiben, wenn dies durch den PaaS Anbieter bereits mitberücksichtigt wird.

Jedoch sind neben Vorteilen auch Nachteile bei der Nutzung einer PaaS Umgebung zu berücksichtigen:

- **Abhängigkeit von Anbieter:** Bei Nutzung besteht eine starke Abhängigkeit zu dem Anbieter. Der spätere Umstieg auf einen anderen Anbieter kann schwierig oder gar nicht möglich sein. Ursächlich hierfür kann der Einsatz einer proprietären Lösung sein oder der eingeschränkte Zugriff auf Daten sein. Dieser Effekt wird als vendor lock-in bezeichnet. Es kann sein, dass vorgenommene Investitionen bei Wechsel zu einem anderen Anbieter erneut erforderlich sind oder bei einem Anbieter geblieben werden muss, obwohl dieser ineffizient ist.
- **Vertrauen:** Eine weitere negative Seite von PaaS Modellen ist das erforderliche Vertrauen in den Anbieter. Der Anbieter muss alle erforderlichen Maßnahmen z.B. hinsichtlich Datenschutzes und Datensicherheit treffen, durch die Entwickler besteht möglicherweise keine Möglichkeit dies zu verifizieren. Es besteht somit die Gefahr, dass sensible Unternehmensinformationen oder personenbezogenen Daten durch den PaaS Dienstleister nicht in der gewünschten Form verarbeitet werden. Es muss ausreichend Vertrauen in den PaaS Dienstleister existieren, um diese Risiken auszuschließen.

Um den Risiken zu begegnen existieren unterschiedliche Ansätze. So wird etwa vorgeschlagen, zunächst Anwendungen, die nicht geschäftskritisch und keine sensiblen Daten beinhalten zu transformieren. Daraus resultierende Erfahrung kann genutzt werden, um bei Folgeprojekten die richtige Entscheidung zu treffen. Sollte der Einsatz

von PaaS Plattformen sich hierbei bereits als falsch herausstellen, ist die Umstellung auf klassischen Entwicklungsansätze bei nicht-kritischen Anwendungen weniger schmerzhaft.

Ein weiterer Lösungsansatz können Low-Code Entwicklungsplattformen darstellen, die sowohl in der Cloud als PaaS, aber auch auf unternehmenseigener Infrastruktur bereitgestellt werden können. Anbieter wie z.B. Oracle Application Express oder OutSystems bieten diese gemischten Bereitstellungsoptionen für die Entwicklungsplattformen an.

3.3 Abgrenzung No-Code Anwendungsplattformen

Neben dem Begriff Low-Code Entwicklungsplattformen ist zuletzt auch die Bezeichnung No-Code Anwendungsplattformen vermehrt, etwa in Marketingmaterialien von Herstellern, vorzufinden. Unter diesem Begriff sollen Plattformen verstanden werden, die die Bereitstellung individueller Anwendungen ohne jegliche Form der Programmierung oder von Quelltext verstehen. Bei No-Code Anwendungsplattformen handelt sich somit um eine noch weitere Verfolgung des Low-Code Ansatzes. Im Unterschied zur Low-Code Entwicklung soll bei der Bereitstellung von No-Code Anwendungen keine Anwendungsentwicklung mehr stattfinden und somit keine Programmierer erforderlich sein.²⁷

3.4 Anwendungszielgruppe

Eine trennscharfe Abgrenzung der Anwenderzielgruppe von Low-Code Entwicklungsplattformen ist nicht möglich, da sich die Produkte wie in Kapitel 3.1 beschrieben teils auch in ihrer Zielgruppe unterscheiden. Deutlich wird jedoch, dass die meisten Produkte auf die Schnittstelle zwischen klassischen IT-Abteilungen und den eigentlichen Fachanwendern abzielen. Dies jedoch in unterschiedliche Ausprägungen bei den Tools. Manche sind eher für Endanwender optimiert, andere erfordern stärkeres IT-Vorwissen der Anwender.

Die wichtigste Zielgruppe der Low-Code-Bewegung ist immer die, die genau dazwischenliegt: IT-kundige Mitarbeiter der Fachbereiche, sowie IT-Spezialisten, die sich

²⁷ Vgl. Augsten, Was ist Low-Code Development?

über viele Jahre hinweg so tief in die Themen der Fachabteilungen eingearbeitet haben, dass sie längst nicht mehr aktiv in der komplexen objektorientierten Programmierung zuhause sind.²⁸

Für diese Gruppe von fach- und zugleich hinreichend IT-kundigen Mitarbeitern steht auch der in diesem Kontext neu eingeführte Begriff des „citizen developer“. Das ist jemand, der in der Fachabteilung zuhause, zugleich aber auch in der Lage ist, im Bedarfsfall mal etwas zu programmieren.²⁹

3.5 Aufbau und Komponenten

In aktuelle Low-Code Entwicklungsplattformen können unterschiedliche Werkzeuge und Komponenten identifiziert werden, die gemeinsam genutzt werden müssen, um eine Geschäftsanwendung zu entwickeln. Die in aktuell gängigen Low-Code Entwicklungsplattformen meist vorzufindenden Komponenten sollen nachfolgend kurz vorgestellt werden und sind in Abbildung 6 dargestellt.

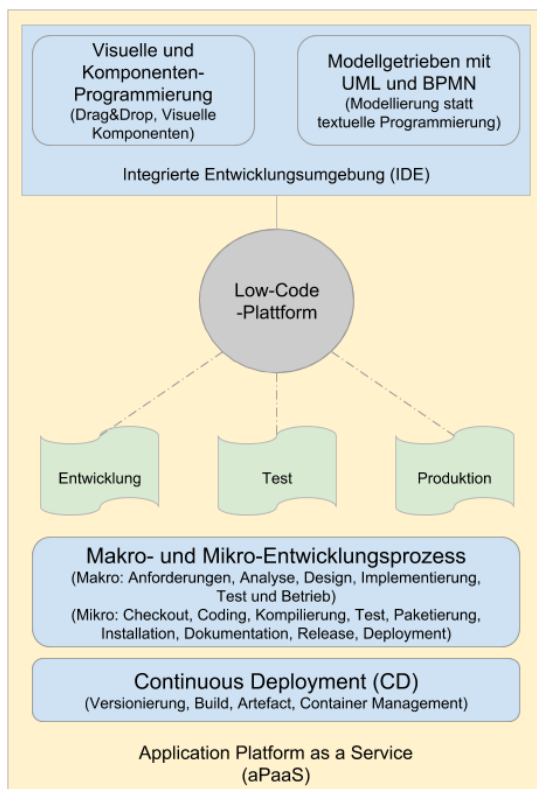


Abbildung 6: Komponenten und Architektur einer Low-Code Entwicklungsplattform (Quelle: Dewanto & Klein)

²⁸ Vgl. Rentrop & Augsten, Low-Code-Entwicklung

²⁹ Vgl. "citizen developer" *Gartner IT Glossary*

Jedoch müssen diese Komponenten nicht in allen Low-Code Entwicklungsplattformen gezwungenermaßen vorhanden sein, so werden diese Werkzeuge teils zusammengefasst oder durch die Plattformen zusätzliche Werkzeuge bereitgestellt.

Datenmodellierungswerkzeug

Die Basis einer Geschäftsanwendung stellt auch in Low-Code Entwicklungsumgebungen meist ein Datenmodell dar. Nach Martin stellt die Erstellung eines fundierten Datenmodells einer der komplexesten Herausforderungen für Nicht-Anwendungsentwickler dar.³⁰

Zur Erstellung des Datenmodells bieten die Low-Code Entwicklungsumgebungen meist eine komfortable Benutzeroberfläche. Mit diesem visuellen Editor können die erforderlichen Datentypen und deren Beziehungen festgelegt werden.

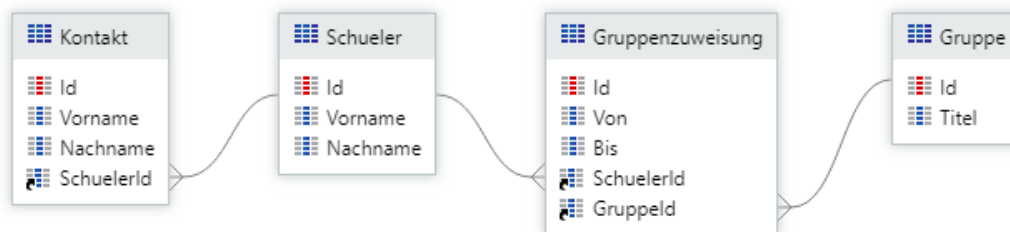


Abbildung 7: Visuelle Datenmodellierung mit OutSystems Low-Code Entwicklungsplattform (Quelle: Eigene Bildschirmkopie)

Einige der Low-Code Plattformen verwenden stattdessen eine vereinfachte Variante, um die Datenmodellierung auch Benutzern aus dem Fachbereich zu ermöglichen. Dabei können Tabellen und alle dazugehörigen Felder per Namen erfasst und die Datentypen ausgewählt werden. Durch die Definition einer Auswahlbox oder einer Datenbeziehung, werden automatisch die relevanten Tabellen und Relationen erstellt, z.B. die Stammdatentabelle für die auswählbaren Werte, oder die Detailtabelle für die Detaildatensätze.

Alternativ wird von Low-Code Entwicklungsplattformen die Möglichkeit angeboten, Tabellenkalkulationsdokumente zu importieren. Dabei wird das Datenmodell auf Grund der Struktur der Tabellenkalkulation abgeleitet und die Daten im Anschluss importiert. Der Charme dieser Lösung besteht darin, dass auch weniger technisch affine Benutzer oft bereits Erfahrung mit der Strukturierung von Daten in Form von Tabellenkalkulationsprogrammen haben. Eine weitere Möglichkeit besteht darin,

³⁰ Vgl. Martin, Application development without programmers, S. 265

dass Datenmodelle aus bereits bestehenden unterschiedlichen Datenbankmanagementsystemen genutzt werden können. Je nach Werkzeug können dabei die Metainformationen der Datenbank, wie z.B. der Datentyp, genutzt werden, um auf dieser Basis geeignete Eingabemasken zu generieren.³¹

Oberflächendesigner

Nachdem ein Datenmodell angelegt wurde, folgt bei den meisten Low-Code Entwicklungsplattformen die Definition der Benutzeroberfläche. Dabei stehen meist verschiedene Vorlagen von häufig verwendeten Anwendungslayouts zur Verfügung, z.B. eine Listenansicht oder eine Detailansicht von Geschäftsdaten. Abhängig vom Werkzeug gibt es hier eine Vielzahl von Möglichkeiten wie z.B. typische leichtgewichtige Web bzw. mobile Listenansichten oder mächtige, editierbare Tabellen für Verwaltungssysteme. Je nach Werkzeug lassen sich diese Vorlagen flexible verändern, um auch große komplexe Applikation mit geringem Aufwand zu erstellen.

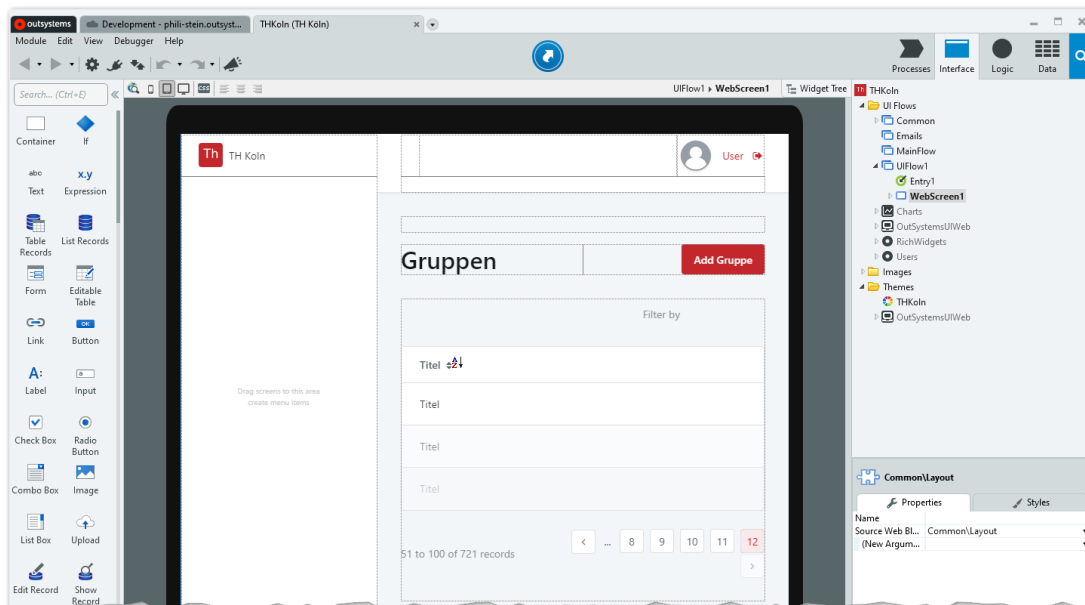


Abbildung 8: Visueller Oberflächendesigner OutSystems Low-Code Entwicklungsplattform (Quelle: Eigene Bildschirmkopie)

Der Oberflächendesigner ermöglicht, dass einzelne Gestaltungselemente per Drag and Drop in die Maske gezogen werden können. Datengebundene Elemente können auch einfach in die Maske gezogen oder mit dem Datenmodell verknüpft werden. Die

³¹ Vgl. Dewanto & Klein, Low-Code == Low Quality?

Elemente der Benutzeroberfläche lassen sich über textuelle oder grafische Eigenschafts-Editoren einstellen. Je nach Werkzeug können auch verschiedenste Layouts zum Positionieren der Elemente gewählt werden.

Es sind zwei grundsätzliche Möglichkeiten zur Definition einer Präsentationsschicht möglich. Dies kann entweder deklarativ durch eine geeignete Auszeichnungssprache (z.B. HTML) oder visuell erfolgen. Da die Benutzerschicht auch ein visuelles Ergebnis darstellt, eignet sich hierbei besonders oft Gestaltung über einen visuellen Editor. Als s.g. WYSIWYG versteht man solche Editoren, die es ermöglichen das Ergebnis visuell zu gestalten. Im Entwicklungsprozess sollte die Benutzeroberfläche z.B. mit visuellen Editoren oder Auszeichnungssprache definiert werden.³²

Die Oberflächendesigner sind oft auch der Ausgangspunkt für die Logiken und Abläufe. Deswegen können meist unterschiedlichste Ereignisse für Gestaltungselemente eingerichtet werden, wie z.B. bei Klick auf eine Schaltfläche das Auslösen einer Geschäftslogik.

Editoren für Geschäftslogik

Auch die Business Logik wird in Low-Code Entwicklungsplattformen häufig visuell erstellt. Dabei gibt es den bekannten Lösungsweg von Workflow Tools, wo durch Flussdiagramme, z.B. unter Einsatz von BPMN aus der modellgetriebenen Softwareentwicklung, die Logiken zusammengesetzt werden.³³

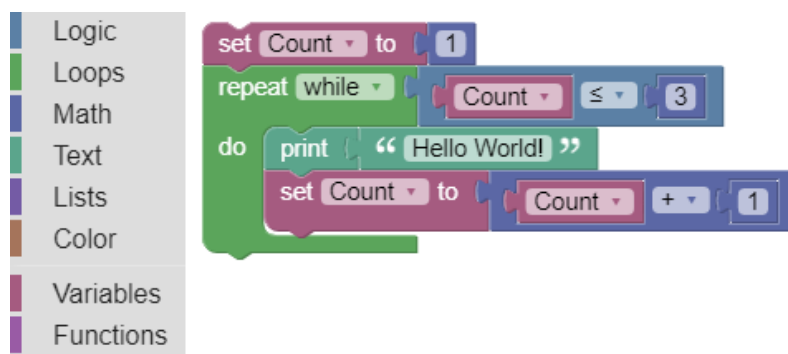


Abbildung 9: Visuelle Ablaufprogrammierung mit Blockly (Quelle: Eigene Bildschirmkopie)

Für Business Logik gibt es jedoch auch kreative neue Ansätze: In Abbildung 9 wird dargestellt, wie beispielsweise mit einem visuellen Editor Ablaufprogramme zusammengesetzt werden, aus denen Geschäftslogik generiert wird.

³² Vgl. Dewanto & Klein, Low-Code == Low Quality?

³³ Vgl. Coutinho, Bringing visual languages to market: The OutSystems story

Verwaltung von Bereitstellung

Eine weitere Komponente nimmt die Bereitstellung der erstellten Geschäftsanwendung z.B. in Form einer Webapplikation ein. Damit verbunden wird von manchen Herstellern auch zusätzliche Funktionalitäten des Life Cycle Managements, wie z.B. Versionsverwaltung- und Ticketmanagement in einer Komponente der Low-Code Entwicklungsplattform bereitgestellt.

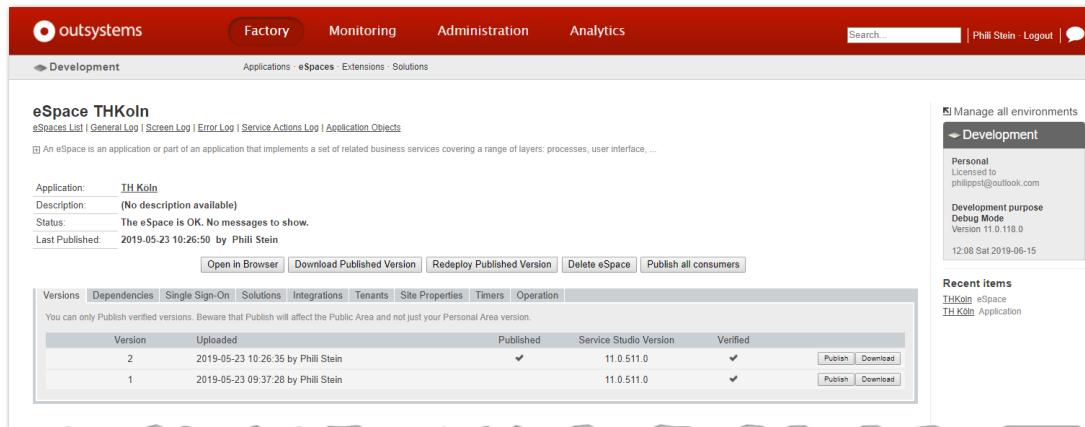


Abbildung 10: Webbasierte Verwaltung von Anwendungsversionen in OutSystems (Quelle: Eigene Bildschirmkopie)

3.6 Marktüberblick

Der aktuelle Markt von Low-Code Entwicklungsplattformen ist sehr dynamisch und schnellwachsend.³⁴ Alle Low-Code Entwicklungsplattformen bieten dabei ähnliche Basisfunktionen zur Datenspeicherung, Abbildung von Geschäftslogiken und Erstellung von Benutzeroberflächen. Eine Segmentierung von Low-Code Entwicklungsplattformen ist deshalb insbesondere auf Basis ihrer Zusatzfunktionen und ihrer spezifischen Anwendungsszenarios möglich. Der nachfolgende Versuch einer Kategorisierung von Low-Code Entwicklungsplattformen orientiert sich dabei an der Einordnung nach Rymer und Koplowitz.

Universelle Lösungen

Die universellen Low-Code Entwicklungsplattformen adressieren ein weites Feld von Geschäftsanwendungen für die Nutzung durch webbasierte und mobile Szenarien. Die Erstellung erfolgt meist mit deklarativen Werkzeugen zur Entwicklung, Integration, Bereitstellung, Lebenszyklusmanagement und Verteilung. Nach Forrester haben die

³⁴ Vgl. Lo Giudice & Condo, Faster Software Delivery Will Accelerate Digital Transformation

Produkte PowerApps der Firma Microsoft, Force.Com der Firma Salesforce sowie OutSystems und Mendix die meiste Marktdurchdringung.³⁵

Prozessorientierte Anwendungsfälle

Low-Code Entwicklungsplattformen für prozessorientierte Anwendungen adressieren insbesondere Anwendungsfälle, in denen die Koordination oder Kollaboration über unterschiedliche Mitarbeiter und Kundenrollen erforderlich ist. Diese Plattformen bieten einen Mix aus Prozessautomatisierung, Vorgangsmanagement und soziale Interaktion Funktionalitäten, die dazu dienen sollen, strukturierte und unstrukturierte Geschäftsprozesse abzubilden. Diese Art von Low-Code Entwicklungsplattformen basieren insbesondere auf Prozess- und Vorgangsmodellierungswerkzeugen. Dabei kommt häufig BPMN zum Einsatz, um visuell Prozessflusslogik und Ad-Hoc Vorgangsmanagement für interne und externe Rollen, Anwendungen und Datenbanksysteme einzurichten. Die meisten Hersteller bieten dabei Metriken, Analysen und Auditing für Prozess- und Vorgangsmonitoring an.

Datenzentrierte Anwendungsfälle

Low-Code Entwicklungsplattformen für datenzentrierte Anwendungsfälle adressieren Geschäftsanwendungen, deren Fokus auf Abfragen, Darstellung und Modifikation von relationalen Daten liegen. Die deklarativen Werkzeuge, die diese Plattformen bereitstellen adressieren die Erstellung von Benutzerinterfaces und Anwendungslogik, Erstellung von Schemata und Datenbanken, Bereitstellung, Datenbindung und Funktionen zur Verwaltung des Lebenszyklus. Als prominenter Vertreter dieses Segments ist Oracle Application Express zu nennen, eine Low-Code Entwicklungsplattform deren Fokus datenzentrierte Anwendungsfälle sind.

Servicemanagement Anwendungsfälle

Als schon ziemlich anwendungsfallspezifisch können die Low-Code Entwicklungsplattformen mit Fokus auf Servicemanagement gesehen werden. Diese Lösungen adressieren Anwendungsfälle in denen die Annahme, Verarbeitung und Nachverfolgung von Serviceprozesse, z.B. durch Kunden- oder Mitarbeiteranfragen im Vordergrund stehen. Ihren Ursprung haben diese Plattformen im Bereich der IT Service Management (ITSM) Lösungen und bieten daher den Entwicklern meist die ITSM Kernfunktionalitäten als konfigurierbare Frameworks mit deklarativen Werkzeugen an. Die

³⁵ Vgl. Rymer & Koplowitz, Low-Code Development Platforms For AD&D Professionals

Werkzeuge, die von diesen Plattformen bereitgestellt werden, sind meist Anwendungserstellung, Workflows, Bereitstellung, Service-Life-Cycle Management und Verteilung. Cherwell Software, KeyedIn Solutions und ServiceNow bieten Low-Code Entwicklungsplattformen in diesem Segment an.

Mobil-native Applikationen

Alle Anbieter von Low-Code Plattformen unterstützen in unterschiedlicher funktioneller Breite und Tiefe die Bereitstellung der Applikationen für mobile Endgeräte. Dies geschieht meist durch die Generierung von webbasierten Benutzeroberflächen, die durch Nutzung von responsiven Gestaltungsstandards z.B. auch für kleine Browserfenster geeignet sind. Jedoch gibt es auch Anbieter, die explizit und ausschließlich den Fokus auf mobile Endgeräte setzen und die Entwicklung von nativen Smartphone Applikationen ermöglichen. Vorteil dieser Lösungen ist, dass diese meist auch den offlinebetrieb vorsehen und Gerätespezifische Funktionen wie z.B. Zugriff auf Kamera ermöglichen. Bei den Anbietern handelt es sich um Appery.io, July Systems, MobileSmith und Snappii.

Neben den vorgestellten Low-Code Entwicklungsplattformen existieren weitere spezifische Lösungen für einzelne Geschäftsfunktionen, wie z.B. elektronische Formulare oder Geschäftsregelmanagement. Diese werden als vertikale Lösungen angesehen und nicht weiter als Low-Code Entwicklungsplattformen in dieser Ausarbeitung berücksichtigt. Gleiches gilt auch für branchenspezifische Lösungen, wie z.B. Bestandsführungssysteme für Versicherungen, die eher als weitreichend konfigurierbare Standardsoftware angesehen werden sollten.

4 Anforderungen an Low-Code Entwicklungsplattform

Um den Anspruch einer ganzheitlichen Plattform zur Entwicklung und Betrieb von Geschäftsanwendungen gerecht zu werden, müssen durch die Low-Code Entwicklungsplattformen unterschiedliche Anforderungen unterstützt und realisiert werden.

Diese Anforderungen sollen in diesem Kapitel gesammelt werden. Im weiteren Verlauf der Arbeit soll die Erfüllung dieser Anforderungen anhand einer prototypischen Realisierung auf Basis einer Low-Code Entwicklungsplattformen bewertet werden.

4.1 Präsentationsschicht

Die Präsentationsschicht dient der visuellen Darstellung von Informationen und der Interaktion des Nutzers mit dem System. Die Präsentationsschicht muss in einer Geschäftsanwendung etwa in der Lage sein, Informationen darzustellen und Benutzer-eingaben entgegennehmen können.

Zeitgemäße Bereitstellungsart

Die Präsentationsschicht sollte einer zeitgemäßen Bereitstellungs- und Anwendungsarchitektur entsprechen. Hierbei gibt es unterschiedliche Arten wie Präsentationsschichten dem Endnutzer zur Verfügung gestellt werden können. Dies kann beispielsweise durch eine auslieferbare Applikation sein, die auf einem lokalen Endgerät ausgeführt werden muss. Alternativ kann die Bereitstellung über eine Webadresse erfolgen, die in einer standardisierten, bereits vorinstallierten Applikation wie z.B. dem Webbrowser ausgeführt wird. Letztere Bereitstellungsoption hat den Vorteil, dass keine anwendungsindividuelle Installation von Software auf den Endgeräten erforderlich ist. Eine weitere Option ist der Einsatz von Plugin-Technologien wie z.B. Flash die bereits auf einigen Rechnern vorinstalliert sind.

Adaptive Benutzeroberfläche

Die Benutzeroberfläche ist in der Praxis in den wenigstens Fällen für alle Nutzer identisch anwendbar. Durch die Heterogenität der Endgeräte (z.B. unterschiedliche Eingabemedien Touch, Tastatur, Maus und Größen, Formate) sollte die Präsentationsschicht adaptiv gestaltet sein. Die Benutzerschicht darf somit nicht nur hart definiert die Positionierung von Gestaltungs- und Eingabeelementen vorsehen, sondern muss auch dynamisch auf Endgeräte reagieren können. Es sollte daher die Möglichkeit bestehen, das Verhalten von Elementen der Präsentationsschicht in Abhängigkeit von diesen Kriterien zu definieren.

Positive Endnutzererfahrung

Die Oberfläche sollte zeitgemäß gestaltet sein und eine positive Benutzererfahrung ermöglichen. Hierbei sollte die Benutzeroberfläche für den Endnutzer funktionell sein, damit dieser effizient seine Aufgaben in der Geschäftsanwendung erfüllen kann. Zugleich sollte sie selbsterklärend, komfortabel und führend auf den Benutzer einwirken.³⁶

Präsentationslogik

Wenngleich der Fokus in der Präsentationsschicht auf der Darstellung der Informationen liegt, kann es erforderlich sein, dass auch in der Präsentationsschicht Anwendungslogiken zu berücksichtigen sind. Dies kann z.B. das Verhalten auf gewisse Interaktionen sein oder Validierung von Dateneingaben durch den Nutzer. Diese Präsentationslogiken sollen ebenfalls durch eine Low-Code Entwicklungsplattform abbildbar sein. Während die rein visuelle Anordnung von Elementen etwa durch einen visuellen Editor realisierbar ist, sollten zusätzlich auch diese Präsentationslogiken berücksichtigt werden können.

4.2 Datenschicht

Die Aufgabe der Datenschicht besteht darin, die im Rahmen der Anwendung erfassten oder generierten Informationen für ihren erforderlichen Zweck vorzuhalten.

Unterschiedliche Informationsarten

Die Anwendung soll in Abhängigkeit vom Anwendungsfall den Umgang mit unterschiedlichen Arten von Informationen ermöglichen. Die Informationen können in Fällen von klassischen Geschäftsanwendungen strukturierte Daten oder auch unstrukturierte Daten sein. Diese können z.B. in Form von Dokumentendateitypen oder in Textform vorliegen. Um aus den Daten Informationen gewinnen zu können, muss neben der Speicherung von Daten auch die Möglichkeit bestehen einen Datenkontext z.B. durch Datenbeziehungen zu speichern.

Datenpersistenz

Die Anwendung soll die Speicherung der Informationen der Geschäftsanwendung sicherstellen. Abhängig vom spezifischen Anwendungsfall sollten darüber hinaus bei

³⁶ Vgl. Jacobs, Modern Enterprise UI Design

der Speicherung von Daten gewisse Eigenschaften wie z.B. Atomarität, Konsistenzerhaltung, Isolation und Dauerhaftigkeit (AKID-Eigenschaften) bei der Verarbeitung von Informationen realisierbar sein.

Datenmodellierung

Um den erforderlichen Kontext von Daten in einer Geschäftsanwendung abzubilden, sollte es eine Möglichkeit geben, Datenstrukturen zu modellieren. Darunter ist insbesondere zu verstehen, aus welchen Informationen, z.B. Feldern und Tabellen, diese sich zusammensetzen und ob die Daten in Zusammenhängen oder Abhängigkeiten bestehen. Bei relationalen Daten sind dies etwa Abhängigkeiten zwischen zwei Tabellen über Fremdschlüssel-Beziehungen. Weiter sollten Dateneinschränkungen (engl. constraints), etwa die Prüfung auf Eindeutigkeit, in der Datenschicht möglich sein.

Datenaustausch und Integration

Häufig reicht es für Anwendungsfälle nicht aus, wenn der Zugriff auf die Daten nur gekapselt über die Anwendungsoberfläche möglich ist. Für den initialen Datenimport, administrative Wartungsaufgaben oder im Falle eines Wechsels der Anwendung kann ein weitergehender Datenzugriff, z.B. unmittelbar auf die Datenbank, erforderlich sein. Die Low-Code Plattform sollte solch einen weitgehenden Datenzugriff ermöglichen.

Anwendungen agieren zunehmend verknüpft mit anderen Systemen und dritten Datenquellen. Die Low-Code Entwicklungsplattform sollte es ermöglichen, auch auf diese Daten des Unternehmens zugreifen zu können. Dies kann im einfachen Fall durch die reine lesende Abfrage von Informationen sein oder in komplexeren Fällen durch schnittstellenbasierte Anbindung von Drittsystemen.

4.3 Geschäftslogik

Die Schicht der Geschäftslogik dient dazu, spezifische fachliche Logiken der Geschäftsanwendung abzubilden. Dabei wirkt diese Geschäftslogik zwischen der Präsentations- und der Datenschicht.

Kalkulation und Aggregation

Auf Basis der Daten aus Benutzerinteraktionen und Datenquellen sollen anwendungsindividuelle Kalkulationen und Aggregationen abgebildet werden können. Das können z.B. anwendungsspezifische Berechnungen, Abfragen oder Eingaben sein.

Lesen und Modifikation von Daten

Gemeinsam mit der Datenschicht sollte die Geschäftslogik in der Lage sein, lesend und modifizierend auf die Daten einwirken zu können. Dies sollte Funktionalitäten zum Lesenden, Erstellenden, Bearbeitenden und Löschenden Datenzugriff beinhalten.³⁷

Datenvalidierungen und Plausibilisierung

Die im Rahmen der Geschäftsanwendung auftretenden Daten sollen über Geschäftslogiken validiert und auf ihre Plausibilität geprüft werden können. Im Zusammenspiel mit der Präsentationsschicht sind dies beispielsweise Validierungen von Benutzereingaben.

Geschäftsprozesse

Um ein Abbild der realen Geschäftslogik in der Geschäftsanwendung zu realisieren, kann es erforderlich sein, eine Menge von verknüpften Arbeitsabläufen und Aufgaben abbilden zu müssen. Diese Geschäftsprozesse zeichnen sich im Regelfall durch ein definiertes Auslöseereignis und dem Erreichen eines gewissen Ergebnisses unter Beachtung bestimmter Regeln und Faktoren dar. Die Low-Code Entwicklungsplattform sollte geeignete Methoden zur Realisierung von Geschäftsprozessen bereitstellen.

Integration externer Geschäftslogik

Es ist davon auszugehen, dass nicht alle Funktionalitäten in der Anwendung selbst umzusetzen sind. Oft kann es erforderlich sein, durch den Aufruf von Drittanwendung Geschäftslogiken abzubilden. Dies steht im engen Zusammenhang mit der externen Datenintegration. Es sollte die Möglichkeit geben, z.B. durch den Aufruf von geeigneten Schnittstellen, externe Geschäftslogik einzubeziehen.

4.4 Querschnitts- und Unterstützungsfunktionen

Neben der Funktionalität die sich der Präsentationsschicht, Datenschicht und Geschäftslogik zuordnen lassen, sollte die Low-Code Entwicklungsplattform noch weitere Anforderungen, häufig als nicht-funktionale Anforderungen bezeichnet, bei der Realisierung von Geschäftsanwendungen unterstützen.

³⁷ Englisch als Create, Reade, Update, Delete (kurz CRUD) Funktionalität beschrieben

Anwendungs- und Datensicherheit

Alle zu verarbeitenden Informationen sowie sämtliche Funktionalitäten der Anwendung müssen unter Berücksichtigung von Anwendungs- und Datensicherheit durch die Low-Code Entwicklungsplattform realisiert werden. Dabei muss die Entwicklungsplattform insbesondere gängige Angriffsmustern und häufig auftretenden Sicherheitsrisiken vorbeugen.

Benutzerauthentifizierung und Autorisierung

Um den Zugriff auf die Anwendung und deren Nutzung zu beschränken, sollten geeignete Funktionalitäten zur Authentifizierung und Autorisierung von Anwendungsnutzern bereitgestellt werden. Die einfachste Form der Authentifizierung kann hierbei aus einem Benutzerverzeichnis in der Anwendung bestehen und der Abfrage eines Geheimnisses, etwa in Form eines Benutzernamens und Passworts. Insbesondere im Kontext von Geschäftsanwendungen, die in Unternehmen eingesetzt werden, ist jedoch vorstellbar, dass die Authentifizierung auf bestehenden Systemen wie einem unternehmensweiten Benutzerverzeichnis aufsetzen muss. Vermehrt kommen in Unternehmen auch Dienste zum Einsatz, die auch den Vorgang der Authentifizierung zentral abbilden. Dies ermöglicht die einmalige und zentralisierte Authentifizierung (engl. single-sign-on). Die Low-Code Entwicklungsplattform sollte somit die Nutzung von bestehenden Benutzerverzeichnissen und Authentifizierungsdiensten ermöglichen.

Kollaborative Anwendungsentwicklung

Auch in einer Low-Code Entwicklungsplattform ist nicht davon auszugehen, dass die Entwicklung einer Anwendung durch eine einzelne Person erfolgt. Die Plattform sollte deshalb die gemeinsame Arbeit von mehreren an der Anwendungsentwicklung beteiligten Personen ermöglichen. Dies kann durch die Bereitstellung von Kollaborationswerkzeugen wie einem Fehlertracker aber auch auf Anwendungsebene, z.B. durch die Sperrung einzelner Anwendungselemente, wenn diese bereits durch einen Nutzer modifiziert werden, realisiert werden.

Änderungsnachverfolgbarkeit

Um Entwicklungsfortschritte und unterschiedliche Varianten der Anwendung differenzieren zu können, sollte in der Low-Code Entwicklungsplattform eine Möglichkeit existieren, diese abzugrenzen. Hierfür kann beispielsweise eine geeignete Anwendungsversionierung eingesetzt werden. Werden Änderungen an der Anwendung vorgenommen, so sollten diese nachvollziehbar dokumentiert werden. Hierbei muss insbeson-

dere der Umfang der Änderung, z.B. welche Felder, Inhalte, Oberflächen, der Zeitpunkt der Änderung und die Person, die die Änderung vorgenommen hat, dokumentiert werden. Diese Angaben sollten es in Kombination mit einer Auftragsnummer ermöglichen transparent zu machen, wie eine Änderung der Anwendung zustande gekommen bzw. durch wen diese veranlasst wurde.

Anwendungsdebugging

Wenngleich die mögliche Anzahl von Fehlerquellen im Entwicklungsprozess durch die Methodiken und begrenzten Einflussmöglichkeiten der Low-Code Entwicklungsplattform vermutlich bereits deutlich geringer sind wie bei der klassischen Softwareentwicklung, sind Fehlersituationen nicht ganz auszuschließen. Zur Eingrenzung und Identifikation der Fehlerursachen sollten Werkzeuge zur Identifizierung und Analyse von Fehlersituationen bereitgestellt werden.

Anwendungstest

Die Anwendung muss im Rahmen der Entwicklung getestet werden können. Durch die deutliche Verringerung des Quelltextes werden hierbei vermutlich von der klassischen Softwareentwicklung abweichende Testansätze und Methoden gewählt. Die Low-Code Entwicklungsplattform sollte den Anwendungstest ermöglichen und darüber hinaus geeignete Werkzeuge zur Testautomatisierung bereitstellen.

Anwendungsbereitstellung

Eine Anwendung muss angemessen für den Betrieb bereitgestellt werden können. Im Unterschied zu anderen Stadien, wie dem Entwicklungs- oder dem Testbetrieb, kann es im Produktivbetrieb erforderlich sein, dass eine größere Anzahl von Nutzern und Lastspitzen hinsichtlich der Zugriffsmenge auftreten. Die Plattform muss Möglichkeiten bereitstellen eine hierfür angemessene Ressourcenskalisierung der Anwendung zu ermöglichen.

Umgebungstrennung

Um die Anwendungsentwicklung auch während des Betriebs zu ermöglichen, muss es Möglichkeiten geben, zwischen unterschiedlichen Systemumgebungen zu differenzieren. Dies kann z.B. durch die Bereitstellung einer eigenständigen Entwicklungsumgebung erfolgen. Erst nach Abschluss aller erforderlichen Anpassungen, Bereinigung von Fehlern und Abschluss des Tests werden die Änderungen gesammelt in eine Produktivumgebung übergeben. Hierauf zählt auch die Anforderung der Anwendungsversionierung ein. Neben der Trennung der Umgebung muss somit auch die

Bereitstellung von neuen Anwendungsversionen von einer in eine andere Systemumgebung ermöglicht werden.

Ereignisprotokollierung

Besondere Ereignisse, die im Laufe bzw. Betrieb der Anwendung auftreten, z.B. Fehler- oder Ausnahmesituationen, sollten durch die Anwendung protokolliert werden. Diese Protokolle können etwa der Identifikation und Analyse von Anwendungsfehlern dienen. Die Protokollierung berücksichtigt hierbei möglichst detaillierte Informationen zu diesem Ereignis, wie z.B. Zeitpunkt und Programmzustände.

4.5 Anforderungskatalog

Aus der Sammlung von Anforderungen an eine Low-Code Entwicklungsplattform ergibt sich der in Tabelle 2 dargestellte Anforderungskatalog. Dabei entspricht die Anforderungsnummer jeweils dem entsprechenden vorangegangenen Kapitel.

Nr.	Bereich	Anforderung
1.1	Präsentationsschicht	Zeitgemäße Bereitstellungsart
1.2	Präsentationsschicht	Adaptive Benutzeroberfläche
1.3	Präsentationsschicht	Positive Endnutzererfahrung
1.4	Präsentationsschicht	Präsentationslogik
2.1	Datenschicht	Unterschiedliche Informationsarten
2.2	Datenschicht	Datenpersistenz
2.3	Datenschicht	Datenmodellierung
2.4	Datenschicht	Datenaustausch und Integration
3.1	Geschäftslogik	Kalkulation und Aggregation
3.2	Geschäftslogik	Lesen und Modifikation von Daten
3.3	Geschäftslogik	Datenvalidierung und Plausibilisierung
3.4	Geschäftslogik	Geschäftsprozesse
3.5	Geschäftslogik	Integration von ext. Geschäftslogik
4.1	Querschnitts- u. Unterstützungsfunktionen	Anwendungs- / Datensicherheit
4.2	Querschnitts- u. Unterstützungsfunktionen	Benutzerauthentifizierung / Autorisierung
4.3	Querschnitts- u. Unterstützungsfunktionen	Kollaborative Anwendungsentwicklung
4.4	Querschnitts- u. Unterstützungsfunktionen	Änderungsnachverfolgbarkeit
4.5	Querschnitts- u. Unterstützungsfunktionen	Anwendungsdebugging
4.6	Querschnitts- u. Unterstützungsfunktionen	Anwendungstest
4.7	Querschnitts- u. Unterstützungsfunktionen	Anwendungsbereitstellung
4.8	Querschnitts- u. Unterstützungsfunktionen	Umgebungstrennung
4.9	Querschnitts- u. Unterstützungsfunktionen	Ereignisprotokollierung

Tabelle 2: Anforderungskatalog für Low-Code Entwicklungsplattformen

5 Prototypische Realisierung

Im folgenden Kapitel soll prototypisch ein Anwendungsfall mit der Low-Code Entwicklungsplattform Oracle Application Express (Oracle APEX) realisiert werden. Dabei soll geprüft werden, ob gängige Anforderungen an die Entwicklung einer Geschäftsanwendung durch die Low-Code Plattform unterstützt und umgesetzt werden können.

5.1 Anwendungsfall

Der Stadtjugendring Wolfsburg e.V. betreibt seit dem Jahr 2015 Ganztages-Betreuungseinrichtungen für Schülerinnen und Schüler der Stadt Wolfsburg. Durch diese Einrichtungen können Eltern eine Betreuung ihrer Kinder im Anschluss an den regulären Schulunterricht sicherstellen. Die Betreuungseinrichtungen sind an die Schulzentren räumlich angegliedert.

Der Betrieb der Betreuungseinrichtungen erfordert spezifische Verwaltungs- und Dokumentationsaufgaben. Dies reicht von der Übersicht der angemeldeten Schüler, über Anwesenheitsberichten bis zur Personalzuordnung zu den Betreuungsgruppen. Dies erfolgt bisher auf Basis von Tabellenkalkulationsdokumenten die monatlich von den Einrichtungen an die Zentrale gesendet und dort zusammengetragen und für Abrechnungen, Statistiken und Meldungen weiterverarbeitet werden. Die manuellen Vorgänge wie Datenkonsolidierung sind teils fehleranfällig und erfordern in hoher Menge personelle Ressourcen. Durch unterschiedliche und ggf. inkonsistenter Daten zwischen den Einrichtungen und der Zentrale können weitere Fehler verursacht werden.

Die relevanten Daten sollen zukünftig in einer zentralen Geschäftsanwendung erfasst werden, auf die alle Mitarbeiter gemäß ihrer Funktionsrolle Zugriff erhalten. Der Austausch von Informationen über Tabellenkalkulationen, per E-Mail oder in Papierform soll entfallen. Die Anwendung soll die Benutzer bei der Erledigung ihrer Aufgaben möglichst unterstützen und zur Vermeidung von Fehlern z.B. durch Einsatz von angemessenen Dialogen, Plausibilitätskontrollen und Validierungen beitragen. Der Zugriff auf die Daten sowie deren Bearbeitung soll möglichst einfach benutzergeschützt über die internen IT-Arbeitsplätze sowie damit verbundenen mobilen Geräten erfolgen können.

Anforderungen an prototypische Anwendung

Im Folgenden sollen auszugsweise relevante funktionale Anforderungen an die Anwendung aufgeführt werden. Dies soll dem besseren Verständnis der Zielanwendung

dienen und hat nicht den Anspruch einer abschließende Fachkonzeption der zu erstellenden Software.

- Definition von Betreuungseinrichtungen: Es muss möglich sein die einzelnen Betreuungseinrichtungen einschließlich Anschrift, Kontaktdaten und Leitung der Einrichtung zu erfassen.
- Definition von Betreuungsgruppen: Je Betreuungseinrichtung existieren mehrere Betreuungsgruppen, eine Struktur ähnlich von Schulklassen.
- Stammdaten zu Schüler/innen erfassen und bearbeiten: Die zu betreuenden Schüler müssen mit umfangreichen Stammdaten wie z.B. Namen, Geburtsjahr, Gesundheitsdaten und Kontaktdaten von Ansprechpartnern (z.B. Erziehungsberechtigte, Notfallkontakte, ...) hinterlegt werden können.
- Schüler zu Betreuungsgruppen zuordnen: Es muss möglich sein für einen Zeitraum einen Schüler einer Betreuungsgruppe zuzuweisen.

Neben den funktionalen Anforderungen gilt es folgende nicht-funktionale Anforderungen bei der Anwendung insbesondere zu berücksichtigen:

- Datenverarbeitung im eigenen Netzwerk: Mit der Anwendung werden personenbezogene und besonders schützenswerte Informationen verarbeitet. Aufgrund gesetzlicher Anforderungen und verbindlicher Vorgaben durch kommunale Behörden, darf die Verarbeitung der Daten ausschließlich in eigenen Rechnernetzwerken auf eigene Infrastruktur erfolgen. Die Datenverarbeitung durch Dritte und der Produktivbetrieb in externen Rechenzentren oder Cloud-Infrastrukturen ist ausgeschlossen.
- Zugriffskontrolle und Berechtigungsrollen: Nur berechtigte Benutzer dürfen Zugriff auf die Informationen des Systems erhalten. Der Zugriff muss weiter darauf beschränkt werden, dass die Nutzer nur auf Datensätze und Funktionalitäten zugreifen können, die für ihre Arbeit erforderlich sind.
- Revisionssicherheit von Änderungen: Änderungen die an Datensätze, insbesondere an Schülerdaten vorgenommen werden, müssen unter Berücksichtigung von Person, die die Änderung vorgenommen hat, dokumentiert werden.

Realisierung mit Low-Code Entwicklungsplattform

In einer vorausgegangenen Analyse wurde festgestellt, dass keine Standardsoftware zur Abbildung der fachlichen Anforderungen an die Geschäftsanwendung existiert. Einige fachlichen Anforderungen könnten durch Software aus dem Bereich der Schulverwaltung abgedeckt werden, Software dieser Art stellte sich jedoch als zu Komplex dar und würde umfangreiche Anpassungen bedingen. Software aus dem Bereich von

Kindergärten oder Kinderbetreuungseinrichtungen decken die fachlichen Anforderungen ebenfalls nicht befriedigend ab. Es wird daher die Eigenentwicklung einer Software angestrebt, die den Anforderungen entspricht. Hinsichtlich der Realisierung sind zusätzlich folgende Kriterien zu berücksichtigen:

- **Kosteneffizienz:** Die Entwicklung sollte möglichst preisgünstig erfolgen, da der Träger der Betreuungseinrichtungen als gemeinnütziger Verein nur bedingt finanzielle Ressourcen aufbringen kann.
- **Zeiteffizienz:** Der Entwicklungszeitraum der Anwendung sollte, nicht nur aus Gründen der Kosteneffizienz, möglichst kurzgehalten werden. Ein Einsatz der neuen Software ist bereits nach drei Monaten geplant.
- **Softwarebetrieb:** Für den Betrieb der Software ist kein Personal vorgesehen. Die Anwendung sollte daher möglichst ohne erforderliche Wartung im Betrieb funktionieren.
- **Erweiterbarkeit:** Durch rechtliche Änderungen oder Erfahrungen aus der Praxis soll es möglich sein, zu einem späteren Zeitpunkt die Anwendung mit geringem Aufwand zu erweitern.

Um diese Anforderungen sinnvoll umzusetzen, wurde entschieden die Entwicklung auf Basis einer Low-Code Entwicklungsplattform durchzuführen.

5.2 Oracle Application Express

Im Fokus der Low-Code Entwicklungsplattform Oracle Application Express (Oracle APEX) steht die Entwicklung von webbasierten Anwendungen die sich durch Lesen und Schreibenden Zugriff auf Datenbanken, s.g. CRUD-Funktionalitäten³⁸ auszeichnen. Oracle APEX kann gemäß des Marktüberblicks und der Segmentierung von Low-Code Entwicklungsplattformen in Kapitel 3.6 als Low-Code Entwicklungsplattform für datenzentrierte Anwendungsfälle eingeordnet werden.

Wie der Produktbezeichnung zu entnehmen ist, handelt es sich um ein Produkt des US-amerikanischer Soft- und Hardwareherstellers Oracle Corporation. Die Oracle Corporation zählt zu den weltweit größten Softwareherstellern und ist insbesondere durch das kommerzielle Datenbankmanagementsystem Oracle Database im Unternehmenseinsatz weit verbreitet.

³⁸ Abkürzung für Create, Read, Update und Delete

Oracle APEX stellt die Anwendungen als webbasierte Applikationen bereit. Die Nutzung ist mit einem Webbrowser möglich. Oracle APEX generiert dynamische HTML-Seiten in Verbindung mit CSS Auszeichnung und JavaScript Quelltext die vom Webbrowser dargestellt werden. Auf einem Client-PC ist somit keine Installation von spezifischer Software erforderlich, jedoch bedarf es einen modernen Webbrowser sowie eine Internetverbindung.³⁹

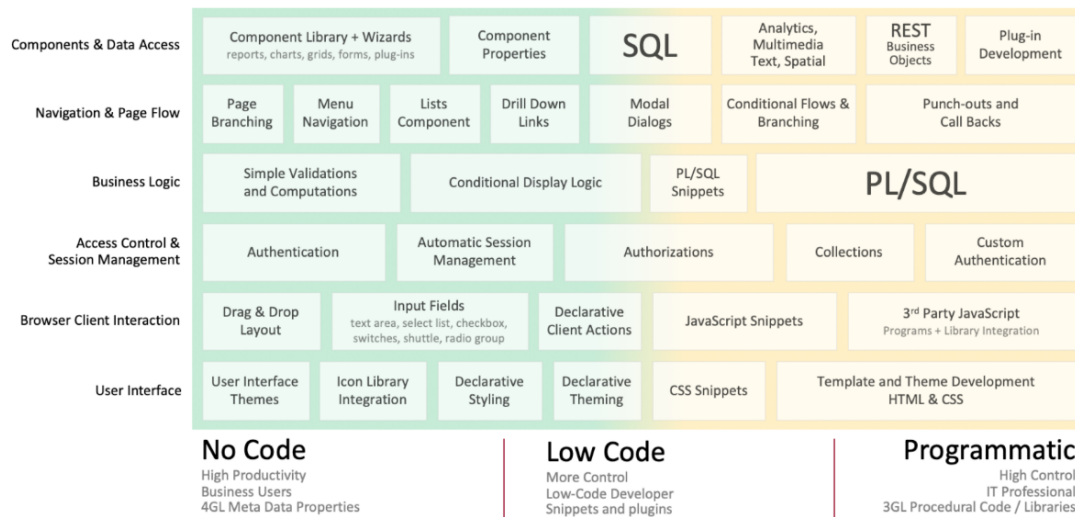


Abbildung 11: Oracle APEX Entwicklungstypen (Quelle: Oracle Corporation)

Laut Selbstbeschreibung des Herstellers handelt es sich bei Oracle APEX um eine Entwicklungsplattform, die dem Low-Code Paradigma folgt.⁴⁰ Durch eine Vielzahl von zusätzlichen Möglichkeiten der Anwendungserweiterung oder der ausschließlichen Nutzung von Standardkomponenten, kann Oracle APEX zusätzlich auch als No-Code Anwendungsplattform oder programmatische Standardanwendung angesehen werden. Der variable Einsatz von mehr oder weniger Quelltext ist in Abbildung 11 ersichtlich.

Historie und Ausblick

Oracle APEX kann auf eine rund 20-jährige Historie zurückblicken. Den Ursprung nahm die Software im Jahr 2000 als die Oracle Entwickler Mike Hichwa und Joel Kallman einen firmeninternen, webbasierten Veranstaltungskalender mit Anbindung an die Unternehmensdatenbank erstellten und hierfür das Framework Flows entwickelten. Das Framework wurde in Folge mit weiteren Funktionalitäten wie z.B. einer

³⁹ Vgl. Jennings, Oracle Application Express, 2.1

⁴⁰ Vgl. Oracle Corporation, Oracle Application Express Website

Verwaltungsoberfläche ausgebaut und für interne Anwendungen der Firma Oracle eingesetzt. Im Jahre 2004 wurde das Framework, zunächst unter dem Namen HTML DB, initial für die Öffentlichkeit freigegeben. Ab dem Jahr wird das Produkt unter dem Namen Oracle Application Express (Oracle APEX) geführt.

Seitdem der initialen Veröffentlichung erfolgt eine kontinuierliche Weiterentwicklung mit Fehlerbehebungen und neuen Funktionalitäten. Es erfolgt etwa jährlich die Veröffentlichung einer neuen Produktversion. Die Oracle APEX Versionsnummern folgen zuletzt jeweils der in Abhängigkeit stehenden Oracle Datenbankmanagementsystem Versionsnummer.

Jahr	Produktbezeichnung	Version	Änderungen / Neue Funktionalitäten
2000	Flows	-	Unternehmensinterner Entwicklungsbeginn
2004	HTML DB	1.5	Initiale Veröffentlichung
2005	HTML DB	2.0	SQL Workshop, Benutzeroberfläche zur Tabellenadministration
2006	Application Express	2.1	Umbenennung zu APEX und Integration in Oracle Express Edition (XE) Datenbank
2006	Application Express	2.2	Möglichkeit Applikationen zu paketieren und in Umgebungen bereitzustellen
2007	Application Express	3.0	Grafische Funktionen und PDF Reports
2008	Application Express	3.1	Interaktive Reports mit JavaScript Technologie
2009	Application Express	3.2	Konvertierung Oracle Forms nach APEX
2010	Application Express	4.0	AJAX Web Funktionalität zum asynchronen Laden von Seitenelementen
2011	Application Express	4.1	Möglichkeit des Dateiuploads und Handling von Datenbanken ROWID
2012	Application Express	4.2	Optimierung für mobile Darstellung mit HTML5
2015	Application Express	5.0	Modale Dialoge und überarbeiteter Seitendesigner
2016	Application Express	5.1	Interaktive Tabellen mit Editierfunktion
2018	Application Express	18.1	Zugriffskontrolle und Monitoring
2018	Application Express	18.2	Optimierung der barrierefreien Darstellung
2019	Application Express	19.1	Integration von externen Webservices

Tabelle 3: Versionshistorie Oracle APEX

Oracle APEX wird von der Oracle Corporation auch selbst eingesetzt, so etwa für die Bereitstellung von Anwendungen für den Kundensupport wie z.B. der Wissensdatenbank AskTom⁴¹ oder dem Trainingsportal LiveSQL⁴².

Neben Oracle APEX existieren noch zwei weitere Produkte im Portfolio der Oracle Corporation, die eine vereinfachte Entwicklung von datenorientierten Anwendungen

⁴¹ Anwendungswebseite AskTom <https://asktom.oracle.com/>

⁴² Anwendungswebseite LiveSQL <https://livesql.oracle.com/>

ermöglichen sollen. Oracle Forms ist eine Rapid Application Development Applikation, um einfache formularbasierte Anwendungen auf Basis der Oracle Datenbank zu erstellen. Die Anwendung wurde initial gemeinsam mit der ersten Version der Oracle Datenbank veröffentlicht und wird seitdem weiterentwickelt. Oracle Forms ist nicht als webbasierte Anwendung ausgerichtet und stellt die Benutzeroberfläche als java-basierte Client-Anwendung bereit.⁴³ Oracle Application Development Framework (Oracle ADF) ist ein Framework zur Erstellung von Anwendungen auf Basis der Java Enterprise Edition Spezifikation. Wenngleich über entsprechende integrierte Entwicklungsumgebungen die Entwicklung von Oberflächen deutlich vereinfacht wird, verfolgt Oracle ADF keinen Low-Code Entwicklungsansatz.

Oracle APEX wird gemäß Entwicklungsplan auch zukünftig aktiv vom Hersteller Oracle Corporation weiterentwickelt und mit zusätzlichen Funktionen ausgestattet. Für das vierte Quartal 2019 wird die Veröffentlichung der Version Oracle APEX 19.2 erwartet.⁴⁴ Während in den Oracle Datenbankversionen bis zum Jahr 2010 Oracle APEX mit der Datenbank mitinstalliert wurde, wird Oracle APEX seit diesem Zeitpunkt als eigenständiges und separates Modul zur Nachinstallation angeboten.

Anwendungsentwicklung

Die gesamte Entwicklung und Administration von Anwendungen der Low-Code Entwicklungsplattform erfolgt über eine webbasierte integrierte Entwicklungsumgebung Oracle APEX IDE. Die Entwicklungsumgebung erfordert keine Installation auf dem Anwenderrechner, da sie über einen Webbrowser aufgerufen werden kann und lediglich eine Authentifizierung mit einem Benutzernamen und Kennwort benötigt.

Lizenzierung

Oracle APEX erfordert keine eigenständige Lizenzierung und kann somit ohne zusätzliche Lizenzierungskosten eingesetzt werden. Oracle APEX wird über die proprietäre Lizenz Oracle Technical Network License bereitgestellt.⁴⁵ Für den Betrieb von Oracle APEX ist jedoch das Datenbankmanagementsystem Oracle Database erforderlich, das im Regelfall mit erheblichen Lizenzgebühren verbunden ist. Der Betrieb mit anderen Datenbankmanagementsystemen wie z.B. der Open Source Datenbank MySQL, ebenfalls vom Hersteller Oracle Corporation, ist nicht möglich.

⁴³ Vgl. Kapitel Rapid Application Development (RAD)

⁴⁴ Vgl. Oracle Corporation, Oracle APEX Statement of Direction

⁴⁵ Vgl. Oracle Corporation, Licensing Information User Manual

Für kleine und mittelgroße Anwendungsfälle kann die kostenfrei erhältliche Version des Datenbankmanagementsystems Oracle Database Express Edition (XE) verwendet werden. Diese Version der Oracle Datenbank unterstützt nur eingeschränkte Ressourcen, etwa die Beschränkung auf maximal 12 Gigabyte Datenbankspeicher, kann jedoch im Rahmen dieser Beschränkungen mit Oracle APEX auch für produktive Anwendungsfälle eingesetzt werden.⁴⁶

Systemarchitektur

Oracle APEX basiert überwiegend auf den Programmiersprachen PL/SQL und JavaScript. Während die gesamte Anwendungslogik einschließlich Seitengenerierung und die Verarbeitung von Hintergrundaufgaben in PL/SQL erfolgt, wird auf der Anwenderseite im Webbrowser JavaScript zur Verarbeitung von Aktionen und Validierungen eingesetzt. Wenngleich die Möglichkeit besteht, auch Java Quelltext zu implementieren, wird diese Programmiersprache von Oracle APEX selbst nicht verwendet.⁴⁷

Im Kern ist Oracle APEX eine Datenbankanwendung und nutzt intensiv Tabellen, Sichten, Indexes, Trigger und Funktionen, die vom Datenbankmanagementsystem bereitgestellt werden. Oracle APEX ist eine deklarative Entwicklungsumgebung, jedes Objekt und jede Konfiguration wird tatsächlich als Metadaten in Datenbanktabellen gespeichert.⁴⁸

Oracle APEX nutzt kaum extern verwalteten Quelltext, sondern legt den gesamten Quelltext in der Datenbank ab und führt diesen auch dort aus.⁴⁹ Wenn eine Seite der Geschäftsanwendung dargestellt werden soll, werden somit auf Ebene der Datenbank PL/SQL Funktionen von Oracle APEX aufgerufen, die auf Basis der in der Datenbank gespeicherten Metadaten sowie der fachlichen Geschäftsdaten, eine Anwendungsseite generieren.

Zum Betrieb der Low-Code Entwicklungsplattform auf eigener Infrastruktur sind folgende Systemkomponente erforderlich:⁵⁰

⁴⁶ Vgl. Oracle Corporation, Licensing Information User Manual

⁴⁷ Vgl. Spendolini, Expert Oracle Application Express Security, S. 26

⁴⁸ Vgl. Spendolini, Expert Oracle Application Express Security, S. 27

⁴⁹ Vgl. Sieben, Oracle APEX, S. 50–51

⁵⁰ Vgl. Sieben, Oracle APEX, S. 53

- **Webbrowser:** Ein Webbrowser dient dem Endanwender zum Zugriff auf die Anwendung. Der Webbrowser muss auf den Endgeräten der Nutzer installiert werden und eine Verbindung mit dem Server, auf dem die Low-Code Entwicklungsplattform bereitgestellt ist, herstellen können.
- **Oracle REST Data Services (ORDS):** Die Anwendung ORDS dient zur Vermittlung von Anfragen zwischen Webbrowser und dem Oracle Datenbankmanagementsystem. Dabei gilt es, die auf dem Internetprotokoll basierenden Anfragen zu interpretieren und in SQL Quelltext zu übersetzen. Antworten des Datenbankmanagementsystems müssen wiederum so bereitgestellt werden, damit diese über das Internetprotokoll transportiert und vom Webbrowser verarbeitet werden können.
- **Oracle Datenbankmanagementsystem:** Das Oracle Datenbankmanagementsystem stellt neben den fachlichen Geschäftsdaten auch die Low-Code Entwicklungsplattform Oracle APEX als Anwendung bereit.

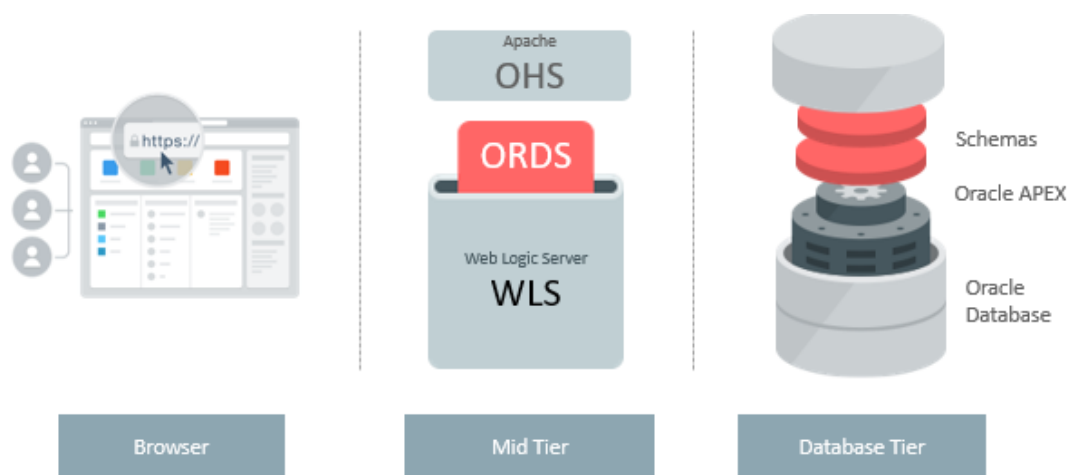


Abbildung 12: Komponenten der Low-Code Entwicklungsplattform Oracle APEX (Quelle: Oracle Corporation)

5.3 Installation Low-Code Entwicklungsplattform

Aufgrund der nicht-funktionalen Anforderungen hinsichtlich des Datenschutzes wurde entschieden, die Low-Code Entwicklungsplattform selbst auf hauseigener Serverinfrastruktur einzusetzen. Ausschlaggebend hierfür war das eine Bereitstellung aufgrund von Datenschutzaspekten nicht in einer Cloud-Infrastruktur möglich ist.

Als Basis für den Prototypen dient ein Intel Xeon Silver Server mit dem Betriebssystem Oracle Linux x86-64 Architektur in der Version 7.6. Der Server wurde in einer virtualisierten Umgebung mit acht Gigabyte Hauptspeicher, 40 Gigabyte Festplattenspeicher und 2.20GHz CPU konfiguriert.

Der Server besitzt keine Benutzeroberfläche, die gesamte Installation und Konfiguration erfolgt daher über die Kommandozeile. Da der Server nicht physikalisch zugänglich ist, erfolgt der Zugriff über das Secure Shell (SSH) Netzwerkprotokoll. Für die Installation und Erstkonfiguration sind weitgehende Benutzerrechte auf dem Server erforderlich. Diese Aufgaben werden daher mit einem Benutzer mit administrativen Funktionsrechten ausgeführt. Der Server ist so vorkonfiguriert, dass dieser unter der Anwendungsdomäne *localhost* adressierbar ist und sämtliche Serverports erreichbar sind.

Die initialen Installations- und Konfigurationsaufgaben gliedern sich in drei Teilschritte:

1. Installation und Konfiguration des Oracle Datenbankmanagementsystems
2. Installation und Konfiguration der Datenzugriffs- und Kommunikationsschicht Oracle ORDS
3. Integration der Oracle APEX Komponenten in zuvor installiertes Datenbankmanagementsystem und ORDS Komponente

Installation Oracle Datenbankmanagementsystem

Die Basis für die Low-Code Entwicklungsplattform stellt das Datenbankmanagementsystem Oracle Database dar, das zunächst installiert werden muss. Im Rahmen der prototypischen Implementierung kommt die Edition Oracle Database Express Edition (Oracle XE) in der Version 18c zum Einsatz. Die Anwendung ist circa drei Gigabyte groß und wird im freien Paketverwaltungsformat RedHat Package Manager (RPM) vom Hersteller bereitgestellt. Abhängig von den verfügbaren Rechnerressourcen ist eine Installationszeit von bis zu 30 Minuten einzuplanen.

Nach der Installation ist eine initiale Konfiguration erforderlich, um mit dem Datenbankmanagementsystem eine Datenbank nutzen zu können. Für die prototypische Realisierung sind keine besonderen Anforderungen hinsichtlich der Konfiguration zu berücksichtigen, aus diesem Grund wird das mitgelieferte Installationsscript für eine Standardkonfiguration mit dem Befehl */etc/init.d/oracle-xe-18c configure* ausgeführt. Es wurden dabei unter anderem folgende Konfigurationen vorgenommen:

- Einrichtung der Containerdatenbank (engl. container database, CDB) mit integrierter Datenbank (engl. pluggable database, PDB) unter Nutzung des lokalen Dateisystems für Datenspeicherung
- Einrichtung der administrativen Benutzer, insbesondere SYS und SYSTEM
- Einrichtung eines Netzwerklistener auf lokalen Host zum SQL Zugriff

- Einrichtung des Administrationswerkzeugs Oracle Enterprise Manager (EM)

Um zu prüfen ob alle Installations- und Konfigurationsschritte erfolgreich verlaufen sind, eignet sich ein Aufruf des Befehls *lsnrctl status*. Durch diesen Befehl wird das Listener Control Utility aufgefordert, den aktuellen Status der Datenbank zu prüfen. Neben Information zum Systemstatus und der eingesetzten Version informiert die Ausgabe über die Endpunkte, unter denen der Netzwerklistener nun Zugriff auf die Datenbank ermöglicht.⁵¹ Als weiterer Prüfschritt bietet sich an, eine Testweise Verbindung über ein SQL Programm, z.B. Oracle SQL Developer, mit der Datenbank aufzunehmen.

Webserver und Dienstschicht Oracle REST Data Services

Die Bereitstellung der Oberfläche sowohl für Nutzung der Low-Code Entwicklungsplattform durch Entwickler als auch für die damit erstellten Geschäftsanwendungen, Bedarf eine webbasierte Zugriffsmöglichkeit. Diese Funktionalität wird durch die Anwendung Oracle REST Data Services (ORDS) bereitgestellt, die als Webserver agiert und Anfragen aus dem Web an die Datenbank übersetzt und Antworten der Datenbank als webbasierte Dokumente zurückgibt. ORDS basiert auf dem quelloffenen in Java entwickelten Apache Tomcat Webserver.

Analog zur Oracle Database wird ORDS als Installationspaket vom Hersteller Oracle Corporation bereitgestellt und muss auf dem eigenen Server lediglich installiert werden. Im Installationsprozess wird abgefragt, über welchen Datenbankbenutzer ORDS mit dem Datenbankmanagementsystem interagieren kann und unter welchen Port der Webserver aus dem Netzwerk erreichbar sein soll.

Die erfolgreiche Installation von ORDS kann geprüft werden, indem über einen beliebigen Webbrowser die Adresse des Webserver mit dem konfigurierten Port 8080, hier also *http://localhost:8080/*, aufgerufen wird. Im Erfolgsfall wird eine initiale Webseite mit dem Hinweis, dass noch keine Anwendung definiert wurde ausgegeben.

Anwendungsmodule Oracle Application Express

Das installierte Datenbankmanagementsystem kann gemeinsam mit der ORDS Dienstschicht bereits für die individuelle Entwicklung von Anwendungen auf Basis der Servertechnologien SQL und PL/SQL genutzt werden. Um jedoch von den Vorteilen

⁵¹ Vgl. Sieben, Oracle PL/SQL, S. 69

einer Low-Code Entwicklungsplattform zu profitieren, ist die Installation der Oracle APEX Komponenten erforderlich.

Oracle APEX kann von der Webseite des Herstellers Oracle Corporation als komprimierte Archivdatei heruntergeladen werden. Diese Archivdatei gliedert sich auf in die Anwendungsbestandteile statische Dateien und PL/SQL Installationsskripte.⁵²

Die statischen Dateien sind Elemente der Webseite wie z.B. Grafiken oder JavaScript Bibliotheken, die stets identisch an jeden Anwender ausgeliefert werden und nicht mit der Datenbank interagieren. Die statischen Dateien müssen in das Verzeichnis für statische Dateien des Webserver ORDS kopiert werden. Anschließend sind diese Dateien über den Webserver abrufbar.

Der dynamische Bestandteil der Low-Code Entwicklungsplattform Oracle APEX wird über die Ausführung des Installationsscripts *apexins.sql* in der Datenbank eingerichtet. Das Installationsscript muss hierfür in einem SQL Konsolenprogramm wie z.B. SqlPlus bei bestehender Datenbankverbindung aufgerufen werden. Die Installation nimmt etwa 10 Minuten in Anspruch und richtet in einem separaten Datenbankschema alle erforderlichen Tabellen und Funktionen zur Nutzung von Oracle APEX ein.

Nach erfolgreicher Installation kann bei erneutem Aufruf der Webserveradresse die Benutzeroberfläche von Oracle APEX eingesehen werden.

5.4 Aufruf und Einrichtung Oracle APEX

Die Mandantenfähigkeit von Oracle APEX erfordert, dass zunächst ein Arbeitsbereich (engl. workspace) eingerichtet wird, in dem die Anwendung entwickelt und bereitgestellt werden kann.

Ein Arbeitsbereich stellt einen isolierten Bereich innerhalb einer Oracle APEX Installation dar. Ein Arbeitsbereich hat eine eigene Benutzerverwaltung und eigene Applikationen. Im Regelfall hat ein Arbeitsbereich zumindest ein eigenes Datenbankschema, kann sich aber Datenbankschema mit anderen Bereichen teilen.

Die Einrichtung des Arbeitsbereichs erfolgt per Zugriff über die Oracle APEX Administrationsoberfläche unter der Webadresse *http://localhost:8080/apex_admin*. Nach

⁵² Download via Webseite <https://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>

Benutzeranmeldung mit dem im Installationsprozess vergebenen administrativen Benutzernamen und Passwort, kann über die Funktion „Arbeitsbereich erstellen“ ein Arbeitsbereich angelegt werden. Folgende Angaben wurden bei Anlage des Arbeitsbereichs angegeben:

- Arbeitsbereich Name: PSTHKOELN
- Datenbankschema Name: PSTHKOELN

Neben den erforderlichen Einstellungen in den Oracle APEX Metadaten wurde mit Anlage des Arbeitsbereiches das Datenbankschema *PSTHKOELN* angelegt. Es handelt sich hierbei um ein leeres, reguläres Datenbankschema im zugrundeliegenden Oracle Datenbankmanagementsystem. Das Schema wird im nachfolgenden Kapitel mit der Datenstruktur und später Geschäftsdaten der Anwendung befüllt.

Nach erfolgreicher Anlage des Arbeitsbereichs, ist nun über die Webadresse *http://localhost:8080/apex* unter Angabe des Namens des Arbeitsbereichs und der Zugangsdaten ein Aufruf der Oracle APEX Entwicklungsoberfläche (engl. Oracle APEX Integrated Development Environment, kurz Oracle APEX IDE) möglich. Die Einstiegsseite der Oracle APEX IDE ist in Abbildung 13 dargestellt.

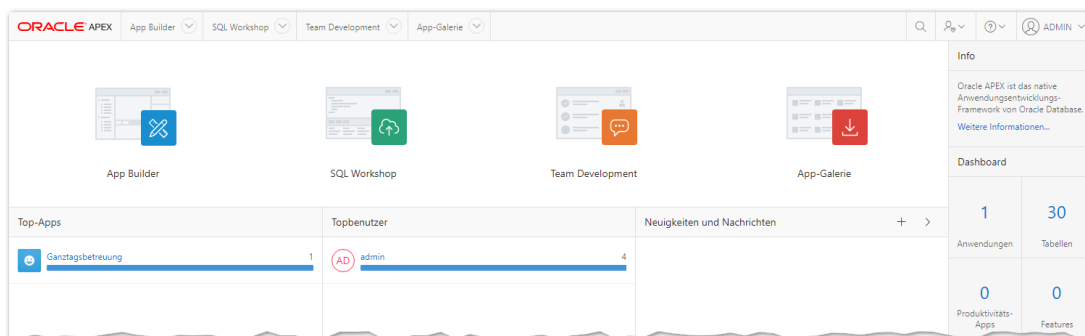


Abbildung 13: Einstiegsseite Oracle APEX IDE (Quelle: Eigene Bildschirmkopie)

Die Oracle APEX IDE ist gegliedert in vier Funktionsbereiche, die zur Entwicklung in der Low-Code Entwicklungsumgebung erforderlich sind:

- Im **Application Builder** (App Builer) werden Funktionen zu Gestaltung, Entwicklung und Entwicklung von Anwendungen aufgerufen. Der überwiegende Teil der Erstellung einer Geschäftsanwendung erfolgt in diesem Funktionsbereich von Oracle APEX.
- Für die Modellierung und Verwaltung von Datenbankobjekten wird mit dem **SQL Workshop** eine Sammlung von Werkzeugen bereitgestellt. Einige dieser webbasierten Werkzeuge sind vergleichbar mit gängigen SQL Editoren wie z.B. Oracle SQL Developer.

- Für die Zusammenarbeit mit mehreren Entwicklern werden im Bereich **Team Development** übergreifende Funktionen bereitgestellt, um die Entwicklung im Team zu organisieren.
- Unter **App-Galerie** wird eine Sammlung von mitgelieferten Anwendungen verstanden, die installiert und eingesetzt werden können. Die bereitgestellten Anwendungen stammen dabei entweder vom Hersteller Oracle Corporation oder individuell vom Unternehmen, dass die Low-Code Entwicklungsplattform einsetzt.

5.5 Datenmodellierung

Die Datenpersistenz von Oracle APEX basiert auf Nutzung des zugrundeliegenden relationalen Datenbankmanagementsystems Oracle Database. Die Oracle Datenbank wird sowohl für die Konfigurations- und Metadaten der Low-Code Entwicklungsplattform als auch für die Modellierung und Persistenz der Geschäftsdaten der entwickelten Anwendungen verwendet.

Um eine klare Trennung zwischen den Daten sicherzustellen, werden unterschiedliche Schemata innerhalb der Datenbank eingesetzt.

- **Schema APEX_190100:** Die Schemata mit Präfix „APEX_“ werden ausschließlich durch die Low-Code Entwicklungsplattform verwaltet. Je Oracle APEX Version wird ein neues Schema angelegt, der Namensteil nach dem Präfix bildet die Version der Plattform ab, hier Version 19.1.0.. Das Schema enthält eine Vielzahl von Tabellen, Sichten, Prozeduren und Funktionen, die die technische Grundlage für Oracle APEX darstellen. Ein SQL Zugriff auf dieses Schema ist nicht erforderlich, da alle erforderlichen Funktionalitäten über die Oracle APEX Entwicklungsoberfläche abgebildet sind.
- **Schema PSTHKOELN:** Je Oracle APEX Arbeitsumgebung wird initial ein leeres Schema für die Speicherung anwendungsspezifischer Geschäftsdaten angelegt.

Für die Modellierung der Anwendungsdaten steht somit ein leeres Schema bereit. Wie die Datenmodellierung erfolgt ist durch die Low-Code Entwicklungsplattform nicht vorgeschrieben, sondern es stehen unterschiedliche Wege und Methoden bereit. Jedoch sollte die Datenmodellierung zunächst erfolgen, da alle weiteren Bestandteile darauf aufbauen in der Entwicklungsplattform angelegt werden.

Zunächst wird ein konzeptionelles Datenmodell entwickelt, dass die fachlichen Anforderungen widerspiegeln soll. Hierzu wurden Entitäten identifiziert, die die spätere Anwendung ausmachen sollen.

Bezeichnung	Beschreibung
Einrichtung	Einrichtung, meist physikalischer Ort, an der die Betreuung von Schülern stattfindet. Eine Einrichtung hat mehrere Gruppen.
Gruppe	Gruppe mit Namen, in der ein oder mehrere Schüler betreut werden. Eine Gruppe ist stets an eine Einrichtung gebunden.
Mitarbeiter	Mitarbeiter/in einer Einrichtung, die mehrere Gruppen bearbeiten kann.
Schüler	Schüler/in der/die in ein oder mehreren Gruppen betreut wird einschließlich Status (Bewerbung, Aktiv, Inaktiv)
Kontakt	Kontakt zu einem Schüler / einer Schülerin, meist Erziehungsberechtigte Personen, z.B. Eltern

Tabelle 4: Datenentitäten der prototypischen Implementierung

Die Beziehungen der Daten werden in Abbildung 14 dargestellt. Um die Übersichtlichkeit zu wahren, wurden die einzelnen Attribute der Entitätstypen nicht berücksichtigt.

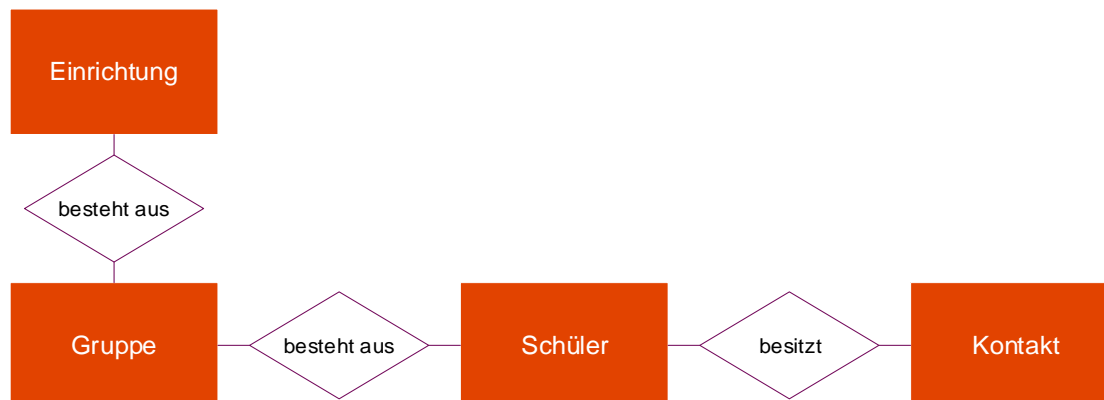


Abbildung 14: Konzeptionelles Datenmodell nach Chen-Notation (Quelle: Eigene Darstellung)

Das Datenmodell soll nun über die Funktion Objektbrowser im Funktionsbereich SQL Workshop der Oracle APEX IDE angelegt werden. Der Objektbrowser ist eine in die Oberfläche integrierte Anwendung zur Entwicklung und Modellierung der Datenschicht. Da keine Installation von Software oder Einrichtung von Zugangsdaten erforderlich ist, ist der Objektbrowser die einfachste Form, um Datenstrukturen und Daten auf der Datenbank anzulegen oder zu modifizieren.

Der Objektbrowser unterstützt den Nutzer bei Anlage oder Modifikation von Tabellen in Form eines Assistenten, der schrittweise die erforderlichen Angaben zu einer Tabelle, wie z.B. Name der Tabelle, Felder mit Datentypen und Fremdschlüssel abfragt und die Änderungen durchführt. Es bedarf hierfür keiner Programmiersprache.

Zunächst soll die Tabelle *Einrichtung* erstellt werden. Hierzu wird die Funktion „Tabelle erstellen“ im Objektbrowser aufgerufen. Die Anlage von Tabellen erfolgt gestützt

durch einen Assistenten in fünf Schritten. Im ersten Schritt des Assistenten wird der Name der gewünschten Tabelle sowie die Spalten einschließlich ihrer Datentypen definiert.

Tabelle erstellen

Spalten

Tabellenname:

☐ Groß-/Kleinschreibung beibehalten

Spaltenname	Typ	Gesamtstellenzahl	Nachkommastellen	Not Null	Identität	Verschieben
<input type="text" value="TITEL"/>	<input type="text" value="VARCHAR2"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="ADRESSE"/>	<input type="text" value="VARCHAR2"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="PLZ"/>	<input type="text" value="VARCHAR2"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="ORT"/>	<input type="text" value="VARCHAR2"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="↑"/> <input type="button" value="↓"/>

Abbildung 15: Oracle APEX Assistent zur Tabellenanlage (Quelle: Eigene Bildschirmkopie)

Im zweiten Schritt des Assistenten kann ein Primärschlüssel für die Tabelle definiert und dessen Generierung, im vorliegenden Fall automatisiert aus einer Sequenz, festgelegt werden.

In Schritt 3 des Assistenten erfolgt die Definition von Fremdschlüsseln. Hierbei wird die Schlüsselspalte der anzulegenden Tabelle sowie die referenzierte Tabelle angegeben. Bei jedem Fremdschlüssel wird das Aktualisierungsverhalten von Assoziationen bei Löschung mit angegeben. Da das Datenmodell keine Fremdschlüssel in der Tabelle *Einrichtung* vorsieht, muss in diesem Schritt keine Einstellung vorgenommen werden.

Im vierten Schritt des Assistenten können Bedingungen für die Tabelle hinterlegt werden. Da in der prototypischen Anwendung ausgeschlossen werden soll, dass versehentlich zwei Einrichtungen mit demselben Namen erfasst werden, wird für die Spalte *TITEL* eine Eindeutigkeits-Bedingung gesetzt.

Der abschließende fünfte Schritt des Assistenten zeigt alle angegebenen Konfigurationen nochmals an und legt bei Bestätigung durch den Nutzer die gewünschte Tabelle an.

Der Assistent muss durch den Entwickler im Anschluss mehrfach durchlaufen werden, bis alle erforderlichen Tabellen für den Prototypen angelegt wurden. Die erfolgreiche Anlage der Tabellen und deren Beziehungen kann anschließend im Objektbrowser über die Modelsicht geprüft und ggf. nachträglich Änderungen vorgenommen werden.

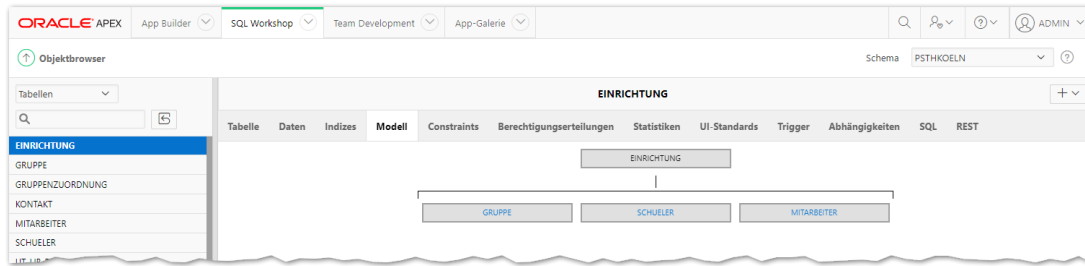


Abbildung 16: Oracle APEX Objektbrowser (Quelle: Eigene Bildschirmkopie)

5.6 Anlage der Anwendung

Nach Erfassung der erforderlichen Datenbankstrukturen im Arbeitsbereich, erfolgt im nächsten Schritt die Anlage der Anwendung. Hierzu muss in der Oracle APEX IDE in den Funktionsbereich App Builder gewechselt werden und der Assistent für die Anlage einer neuen Anwendung aufgerufen werden. Als Name für die Anwendung wird „Ganztagsbetreuung“ gewählt. Die Standardvorbelegungen des Assistenten können beibehalten werden. Nach Abschluss des Assistenten wird die Anwendung in der Page Designer Ansicht der Oracle APEX IDE dargestellt.

Um die erforderlichen Funktionalitäten in der Anwendung abzubilden, wurde auf Basis der fachlichen Anforderung ein Seitenbaum konzipiert. Die Seite „Startseite“ wurde durch den Assistenten bereits automatisch angelegt.

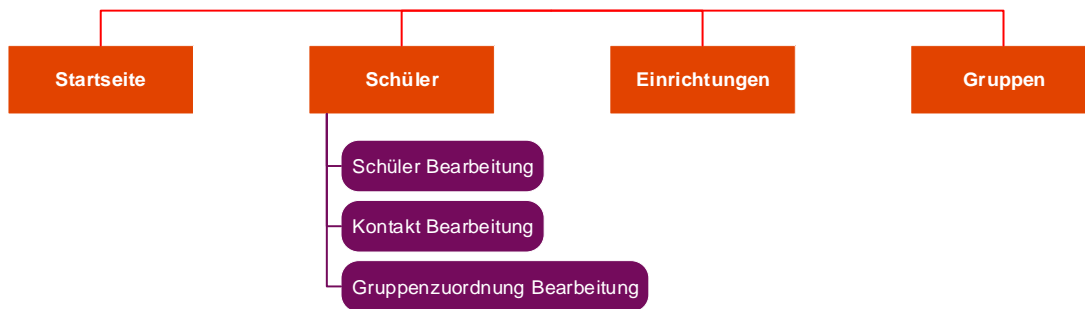


Abbildung 17: Seitenstruktur der prototypischen Anwendung (Quelle: Eigene Darstellung)

Zunächst soll die Seite „Schüler“ angelegt werden. Gewünscht ist, dass diese Seite eine Namensliste aller Schüler darstellt. Bei Klick auf einen Schüler sollen Details zu diesem Schüler wie z.B. das Geburtsdatum, die Kontaktpersonen und die Gruppenzuordnungen des Schülers angezeigt werden. Zusätzlich soll die Möglichkeit bestehen über Dialogfenster neue Schüler zu erfassen und bestehende Schülerdaten zu bearbeiten.

Die Anlage der Seite erfolgt im Page Designer per Klick auf „Seite erstellen“. Als Seitentyp muss „Master-Detail“ ausgewählt, um die gewünschte Darstellung einer Da-

tensatzliste einschließlich Detailansicht zu erhalten. Der Assistent fragt auf der Folgeseite den Seitennamen sowie die zugrundeliegenden Datenbanktabellen ab. Nach Abschluss des Assistenten wird die erstellte Seite im Oracle APEX Page Designer aufgerufen.

Der Oracle APEX Page Designer ermöglicht die Gestaltung einer Seite durch Definition von Seiteneigenschaften, Darstellungs- und Steuerungselementen. Die Benutzeroberfläche des Editors ist, wie in Bildschirmkopie Abbildung 18 ersichtlich, in drei horizontale Spalten aufgeteilt:

- Die linke Spalte stellt die Komponente der Seite in einer Baumstruktur dar. Diese Baumstruktur gibt hierarchische Beziehungen und die Verschachtelungen der Komponenten wieder.
- In der mittleren Spalte werden die Komponenten der Seite in einer Layoutansicht dargestellt.
- Die rechte Spalte ermöglicht, abhängig von der in Layoutansicht angewählten Komponente, die Konfiguration von Eigenschaften.

In allen Spalten können per Drag & Drop Komponenten einfach verschoben und eingefügt werden. Der Page Designer bietet umfangreiche Gestaltungs- und Konfigurationsmöglichkeiten für jede Seite. Neben der Änderung der Anordnung von Komponenten auf einer Seite, können Konfigurationen wie z.B. Verlinkungen vorgenommen werden.

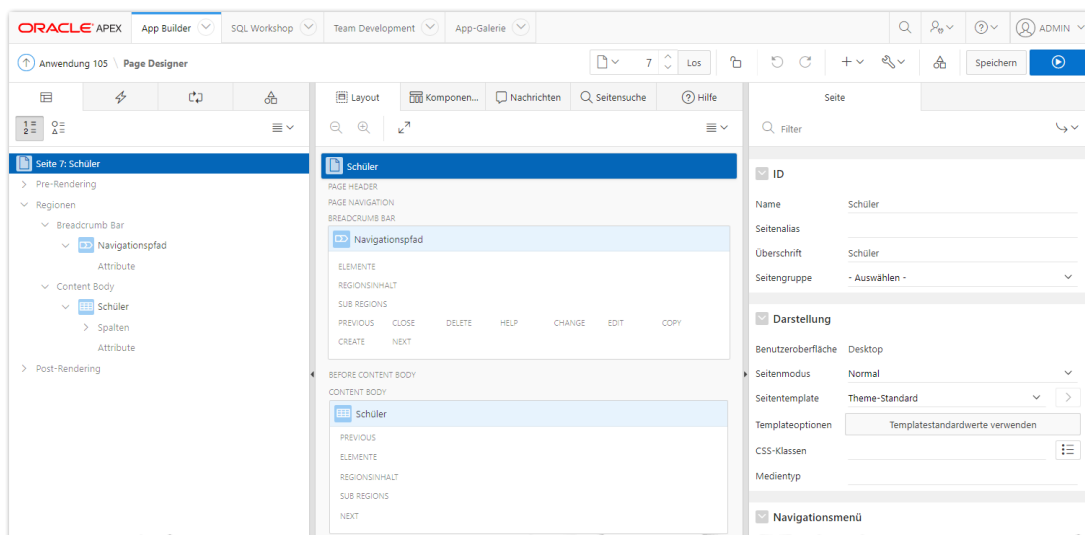


Abbildung 18: Oracle APEX Page Designer (Quelle: Eigene Bildschirmkopie)

An dieser Stelle müssen nur kleinere optische Änderungen vorgenommen werden, so soll auf der Detailansicht eines Schülers nicht die technische ID erscheinen und

anstatt des Fremdschlüssels der Einrichtung, direkt der Titel der Einrichtung. Nachdem alle gewünschten Änderungen im Page Designer vorgenommen wurden, kann per Klick auf „Seite speichern und ausführen“ die tatsächliche Darstellung der Seite im Browser betrachtet werden.

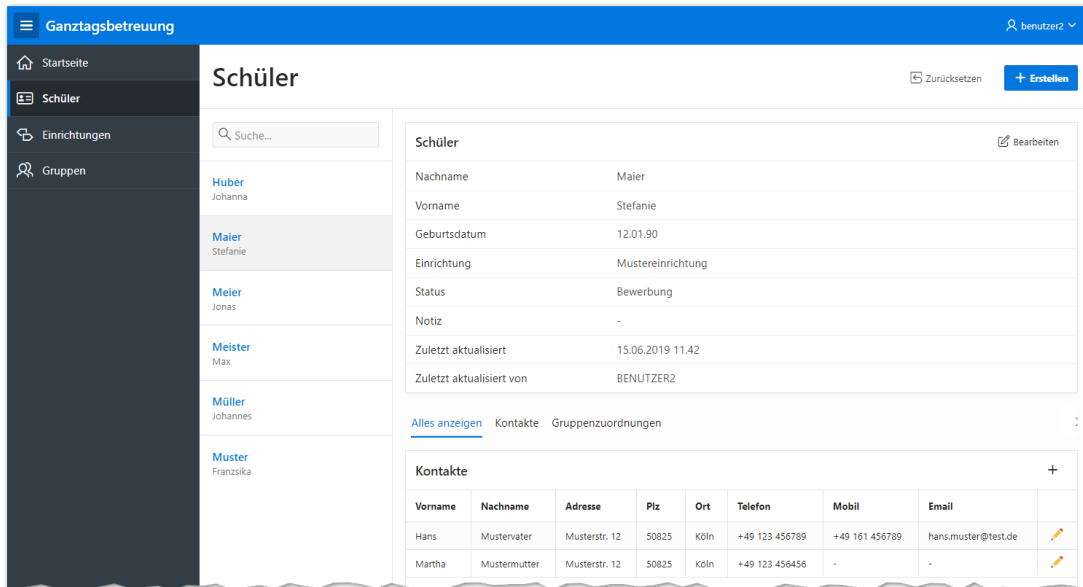


Abbildung 19: Seite Schüler (Quelle: Eigene Bildschirmkopie)

In Abbildung 19 ist zu sehen, wie durch Oracle APEX eine Seite zur Darstellung der Schülerdatensätze generiert wurde. Wie erwartet, wird bei Klick auf einen Schüler, in der rechten Fensterhälfte dessen Details sowie Kontaktdaten und Gruppenzuordnungen dargestellt. Zusätzlich wird bei Klick auf die Schaltflächen „Bearbeiten“ und „Erstellen“ ein Dialogfenster geöffnet, dass die Erfassung der Daten ermöglicht.

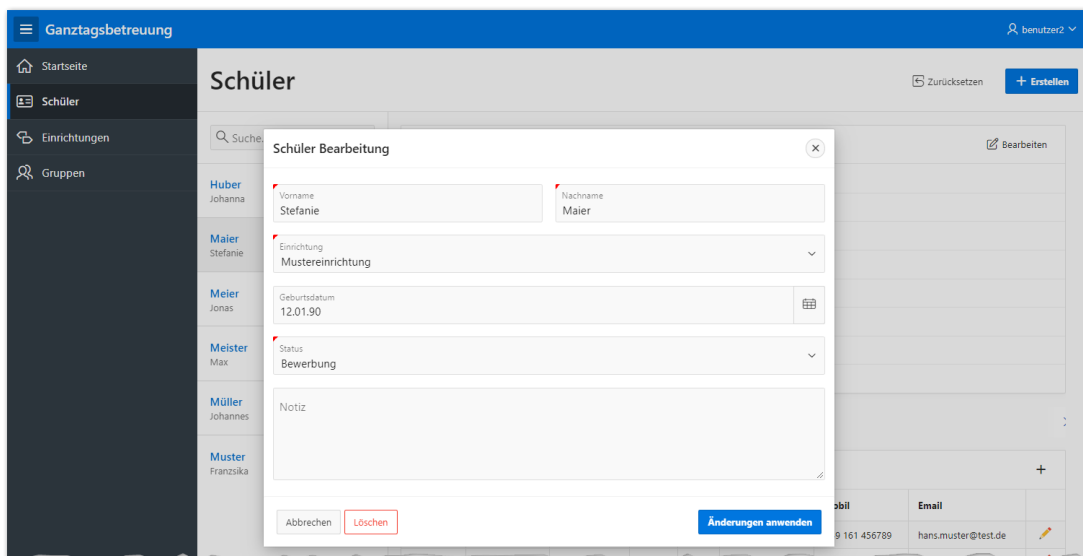


Abbildung 20: Seite Schüler mit geöffneten Dialogfenster (Quelle: Eigene Bildschirmkopie)

Bisher ist es jedoch noch nicht möglich, Einrichtungen und Gruppen über die Benutzeroberfläche zu erstellen oder zu ändern. Um diese Funktionalität zu ergänzen wird erneut der Assistent „Seite erstellen“ im Page Designer aufgerufen. Als Seitentyp wird nun „Interaktives Grid“ ausgewählt. Bei einem interaktiven Grid handelt es sich um eine tabellarische Darstellung der Datensätze, die per Doppelklick in der Tabelle auch direkt Änderungen und Neuanlagen zulässt. Sowohl für die Tabelle *EINRICHTUNG* als auch *GRUPPE* wird jeweils eine Seite angelegt.

	Titel	Adresse	Plz	Ort
<input checked="" type="checkbox"/>	Mustereinrichtung	Musterstr. 12	50825	Köln
<input type="checkbox"/>	Mustereinrichtung 2	Musterstr. 14	50558	Köln

Abbildung 21: Seite Einrichtungen mit interaktivem Grid für Neuanlage und Bearbeitung (Quelle: Eigene Bildschirmkopie)

Die erforderlichen Seiten zur Ansicht und Bearbeitung der Datensätze wurden somit in Oracle APEX angelegt. In den folgenden Abschnitten soll deren Gestaltung und Layout weiter den Bedürfnissen der Geschäftsanwendung angepasst werden.

5.7 Seitengestaltung

Gewünscht ist die Bereitstellung der Anwendungsfunktionen über eine komfortable Benutzeroberfläche, um eine positive Nutzererfahrung zu erzielen. Oracle APEX unterteilt bei der Oberflächengestaltung zwischen dem allgemeinen Seitendesign und der individuellen Gestaltung einer einzelnen Seite.

Unter dem Seitenthema wird die Darstellungsform und das Design von Anwendungen verstanden. Ein Seitendesign ist wiederverwendbar und für unterschiedliche Anwendungen nutzbar. Das Seitendesign beinhaltet keine fachlichen Funktionen, sondern gibt nur den gestalterischen Rahmen vor. Das Seitenthema definiert an welcher Stelle welche Komponentenarten, z.B. im Menü im Kopfbereich der Seite, zugelassen sind und deren Erscheinungsbild.

In der Seitengestaltung werden konkrete Bedienelemente und Regionen abhängig von den fachlichen Anforderungen im Oberflächenlayout platziert. Den Rahmen hierfür gibt das Seitenthema vor, das definiert an welcher Stelle welche Bedienelemente zugelassen sind und wie diese dargestellt werden.

Seitentheme

Eine APEX Seitentheme ist ein auf HTML 5 Technologie basierendes Grundgerüst in Verbindung mit Cascading Style Sheet (CSS) Definitionen einer Seite. Das Theme kann von Grund auf selbstentwickelt werden, ein Standardtheme modifiziert oder ein Standardtheme verwendet werden.

Mit der Anlage der Anwendung wurde das Oracle APEX Standardtheme „APEX Universal Theme“ für die Anwendung vorausgewählt. Es handelt sich hierbei um ein mitgeliefertes Theme das aktuell gängige Anforderungen erfüllen soll und für die responsive Nutzung durch Desktop- und Mobilgeräte ausgelegt ist.

Das Seitengerüst sieht eine Hauptnavigation am linken Bildschirmrand, eine Metanavigation am oberen Seitenrand und einen Inhaltsbereich vor. Der Inhaltsbereich ist aufgeteilt in eine Titelzeile und Gestaltungsraum. Eine ausführliche Beschreibung des Theme und Demonstration ist auf der Webseite des Herstellers vorzufinden.⁵³

Damit die Anwendung von den Nutzern angenommen wird, soll diese das Erscheinungsbild (Corporate Design) aufgreifen. Hierzu sollen insbesondere folgende Änderungen berücksichtigt werden:

- Die Primärfarbe der Anwendung, aktuell ein Standardblauton, soll durch einen Kundenfarbton (grün) ersetzt werden.
- Die Typografie der Anwendung soll durch einen Kundenstandard ersetzt werden

Die Anpassung dieser Konfigurationen erfolgt in Oracle APEX über die Funktionalität Theme Roller. Hierbei handelt es sich um einen Dialog der für Benutzer mit administrativen Berechtigungen unmittelbar in der Oberfläche eingeblendet wird und die Änderung aller Standardwerte des Seitentheme ermöglicht. Die Auswirkung der Änderung eines Konfigurationswertes ist unmittelbar sichtbar und erfordert keine Programmierung.

⁵³ Oracle APEX Universal Theme Referenzdokumentation: <https://apex.oracle.com/ut/>

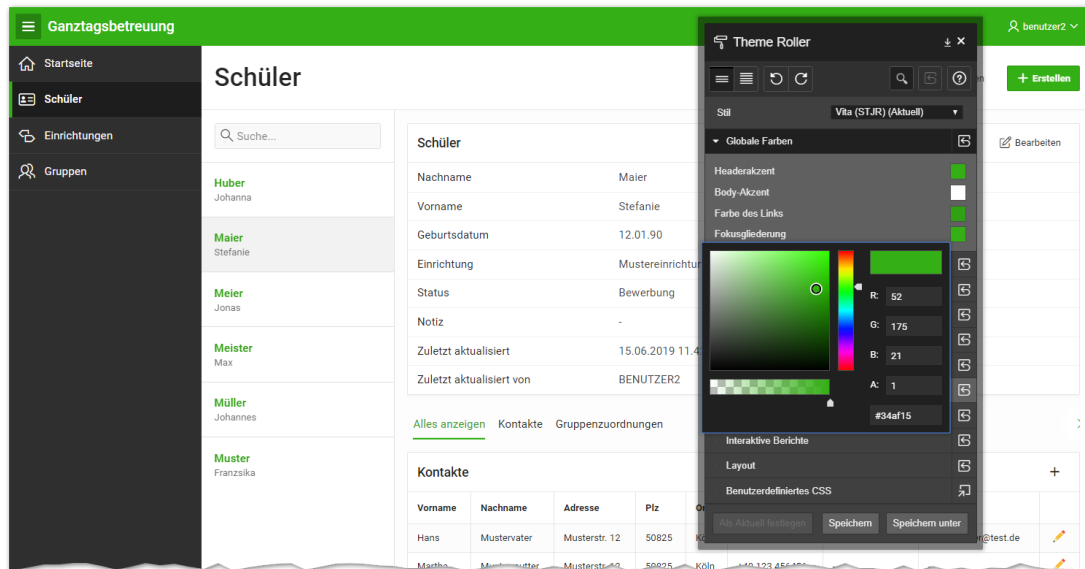


Abbildung 22: Anpassung Seitentheme mit Oracle APEX Theme Roller (Quelle: Eigene Bildschirmkopie)

Seitengestaltung

Die Erstellung der Komponenten einer Seite und ihre Anordnung ist initial durch den Assistenten „Seite erstellen“ erfolgt. Hierbei werden alle im Assistenten angewählten Datenfelder durch einen ihrem Datentyp angemessenen Seitenkomponente dargestellt.

Im Dialog zur Erfassung und Bearbeitung eines Schülers wurden durch Oracle APEX entsprechende HTML Formularelemente gemäß ihren Datentypen im Datenbankmanagementsystem angelegt. In Tabelle 5 ist erkenntlich, dass Oracle APEX hierbei auch die Bedingung berücksichtigt, ob das Datenfeld einen Nullwert annehmen darf und definiert die Formularfelder entsprechend als obligatorische Felder, wenn dies nicht der Fall ist. Weiter werden Fremdschlüssel wie hier die Angabe einer Einrichtung als Auswahlfeld aufgelöst.

Tabelle und Spalte	DBMS Datentyp	HTML Formularelement	HTML Pflichtfeld
SCHUELER.VORNAME	varchar2(50)	input type="text"	Ja
SCHUELER.NACHNAME	varchar2(50)	input type="text"	Ja
SCHUELER.GEBURTSDATUM	date (nullable)	input type="date"	Nein
SCHUELER.NOTIZ	varchar2(255) (nullable)	textarea	Nein
SCHUELER.EINRICHTUNG_ID	number	select	Ja
SCHUELER.STATUS	number	select	Ja

Tabelle 5: Datenmapping Dialog Schüler Bearbeiten

Die Anordnung der Formularfelder erfolgt linear gemäß der Anwahl im Assistenten bei Nutzung der vollen Dialogbreite. Um eine optisch ansprechende Benutzeroberfläche zu erhalten, sollen jedoch fachlich in Verbindung stehende Formularelemente, wie z.B. Vorname und Nachname, in einer Zeile dargestellt werden.

Die Anordnung von Seitenelementen wie hier Formularfelder sowie deren Darstellungsform und Eigenschaften können über den Oracle APEX Page Designer bearbeitet werden. In der folgenden Abbildung ist ersichtlich, wie per Drag & Drop die Formularfelder im Page Designer angeordnet und in der Geschäftsanwendung dargestellt werden.

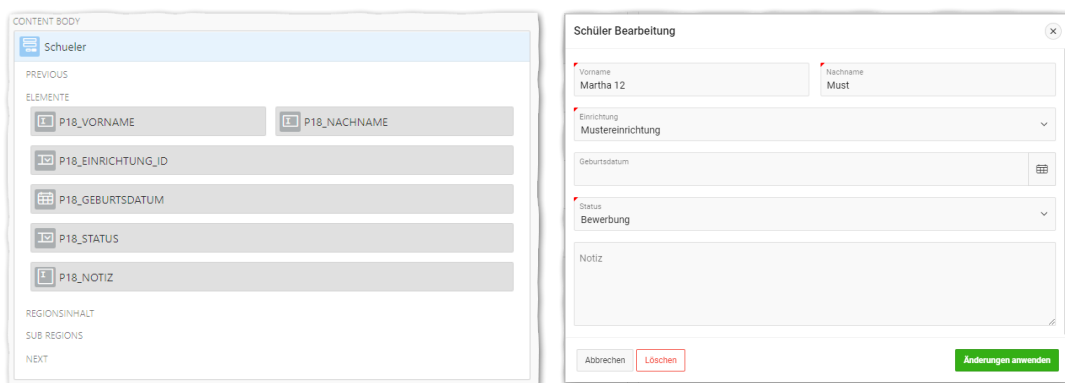


Abbildung 23: Gegenüberstellung Formular in Oracle APEX Page Designer und Ausgabe (Quelle: Eigene Bildschirmkopie)

5.8 Anwendungslogik

In den vorangegangenen Arbeitsschritten wurde neben dem Datenmodell durch die Oracle APEX Assistenten Anwendungsseiten generiert. Bei Betrachtung der Anwendung ist festzustellen, dass bereits durch den Assistenten Anwendungslogik wie z.B. der Sprungmöglichkeiten zwischen den Seiten per Hyperlinks und der Möglichkeit zur Datenerfassung gegeben ist. Diese Anwendungslogik soll nun um anwendungsspezifische Funktionalitäten erweitert werden.

Interaktion innerhalb Anwendung

Da die Geschäftsanwendung durch Oracle APEX als webbasierte HTML Anwendung realisiert wird, erfolgt der Wechsel zwischen den einzelnen Seiten der Anwendung durch Hyperlinks. Mit der Anlage von Seiten in der Geschäftsanwendung über den Assistenten wurde durch Oracle APEX ein Navigationsmenü automatisch generiert und wird im linken Fensterbereich der Anwendung dargestellt wie in Abbildung 22 ersichtlich. Durch Mausklick auf die Menüelemente wechselt der Benutzer zu den jeweiligen Seiten der Anwendung. Die aktuell aktive Seite wird dabei farblich hinterlegt. Ein Wechsel zwischen den Seiten ist jederzeit möglich.

Neben der manuellen Ausführung von Hyperlinks kann auch ein automatischer Aufruf einer Seite sinnvoll sein. In der prototypischen Anwendung soll nach der Anlage eines Schülers, dieser Datensatz unmittelbar angezeigt werden. Um den automatischen Aufruf einer Seite zu realisieren, muss im Oracle APEX Page Designer zunächst die Seite ausgewählt werden, von der die Aktion initiiert wird, in diesem Fall die Dialogseite „Schüler bearbeiten“. Im Abschnitt „Verarbeitung“ wird eine Verzweigung mit der Ausführungsoption „Nach der Verarbeitung“ erstellt. Als Ziel wird die Seite Schüler, hier adressierbar über Seiten ID 17, mit der eindeutigen ID des neu erfassten Schülers als Aufrufparameter übergeben.

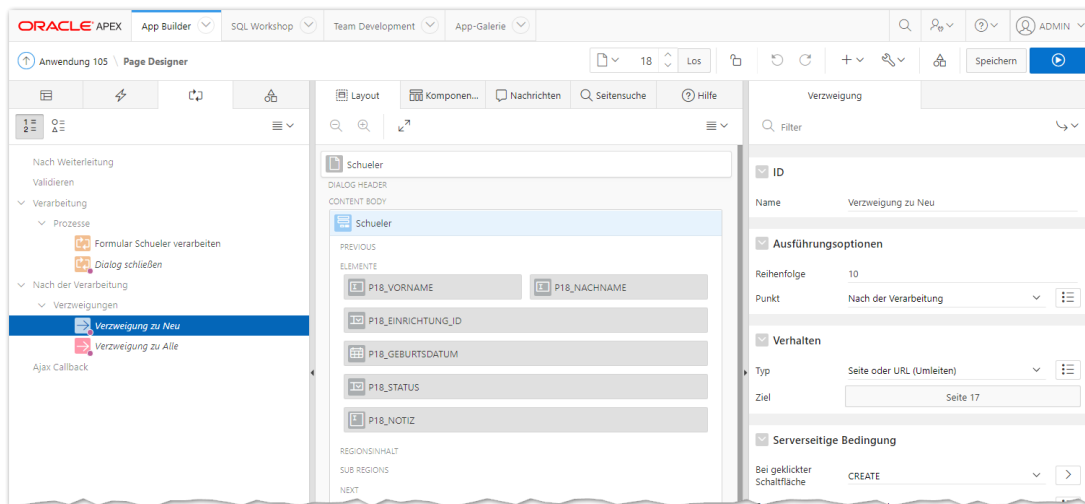


Abbildung 24: Automatische Verzweigung der Seite nach Anlage eines neuen Datensatzes (Quelle: Eigene Bildschirmkopie)

Durch Oracle APEX wird nun automatisch nach jeder Anlage eines Schülers über die Anwendungsoberfläche die Ausführung der Verzweigung angestoßen und der Benutzer somit unmittelbar auf die Detailansicht des neu erfassten Datensatzes geleitet.

Validierung von Benutzereingaben

Bei der Umsetzung der Anforderung ist es Bestandteil der Anwendungslogik, Dateneingaben zu validieren. Nur wenn die Validierung erfolgreich war, dürfen die Anwendereingaben weiterverarbeitet werden. Sollte die Eingabe nicht den Validierungskriterien entsprechen, benötigt der Benutzer einen entsprechenden Fehlerhinweis.

Im Rahmen der prototypischen Implementierung wurden obligatorische Datenfelder als Non-Nullable angelegt und durch Oracle APEX automatisch eine benutzerfreundliche Validierung dieser Felder auf der Oberfläche berücksichtigt. Zusätzlich soll sichergestellt sein, dass ein Schüler, zu einem gewissen Zeitpunkt stets maximal einer Betreuungsgruppe zugeordnet ist. Da der Schüler jedoch auch im Laufe des Schul-

jahres die Gruppe möglicherweise wechseln muss, ist die Beziehung zwischen Schüler und Gruppe als n:m Beziehung über die Datenbanktabelle Gruppenzuordnung gelöst.

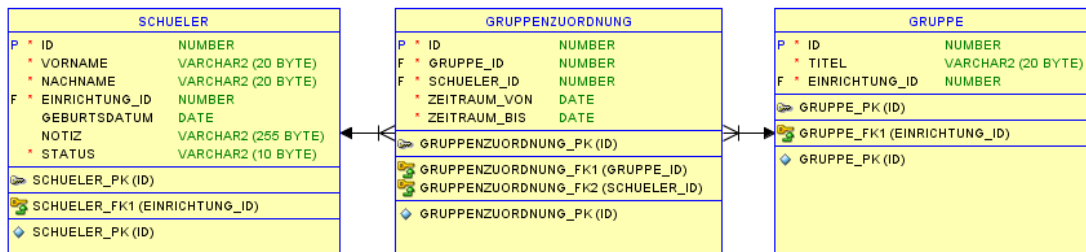


Abbildung 25: Gruppenzuordnung von Schüler zu Gruppe in relationalem Datenmodell (Quelle: Eigene Darstellung)

Um sicherzustellen, dass zu keinem Zeitpunkt ein Schüler gleichzeitig in mehreren Betreuungsgruppe ist, müssen die Feld „Zeitraum von“ und „Zeitraum bis“ dahin gehend durch die Anwendung validiert werden, dass eine Überschneidung durch andere Datensätze ausgeschlossen werden kann. Oracle APEX stellt für diese komplexere Form der Validierung die Möglichkeit bereit, individuelle Datenbankabfragen zu erstellen. Über den Abfragetyp „No Rows returned“ wird Oracle APEX angewiesen, im Falle einer Datenrückgabe der Abfrage, einen Validierungsfehler zu erzeugen und die Anfrage abzulehnen.

```
SELECT z.id
FROM
  GRUPPENZUORDNUNG z
WHERE
  z.schueler_id = :P20_SCHUELER_ID
  AND (
    (:P20_ZEITRAUM_VON BETWEEN z.ZEITRAUM_VON AND z.ZEITRAUM_BIS)
    OR (:P20_ZEITRAUM_BIS BETWEEN z.ZEITRAUM_VON AND z.ZEITRAUM_BIS)
  )
```

Wie in Programmcode erkenntlich, nutzt die Abfrage zur Validierung die Funktionalität des Databindings in der Doppelpunkt Syntax (z.B. :P20_SCHUELER_ID). Dadurch werden die Benutzereingaben aus dem Formularfeldern in der Abfrage ergänzt. Die Abfrage gibt all die Gruppenzuweisungen eines Schülers zurück, deren Gültigkeitszeitraum sich mit der neuen Eingabe überschneidet.

Die zurückgegebene Projektion ist irrelevant, da für die Validierung der Abfragetyp „No Rows Returned“ gewählt wurde, der Inhalt der zurück gegebenen Zeilen aber nicht in die Validierung einfließt. Sollte also durch diese Abfrage eine oder mehrere Zeilen zurückgegeben werden, liegt eine Überschneidung des zu erfassenden Datensatzes mit einem Bestandsdatensatz vor, die Änderung wird somit durch Oracle APEX abgelehnt und ein Hinweis an den Nutzer ausgegeben.

Anmeldestatus von Schülern

Im Rahmen der prototypischen Implementierung ist es erforderlich, den Anmeldestatus von Schülern zu dokumentieren. Nach der Ersterfassung bleiben Schüler so lange im Zustand „Bewerbung“ bis tatsächlich eine Gruppenzuweisung erfolgt ist und der Schüler in den Zustand „Aktiv“ versetzt wurde. Nach Ausscheiden aus der Ganztagesbetreuung kann der Schüler als „Inaktiv“ gesetzt werden.

Die Realisierung erfolgt durch die Anlage eines Feldes *STATUS* in der Schüler Tabelle, dass ausschließlich die Ausprägungen *B* für Bewerbung, *A* für Aktiv und *I* für Inaktiv zulässt. Ein Datenbank Check-Constraint in Verbindung damit, dass das Feld keine Nullwerte zulässt, stellt sicher, dass ausschließlich diese Ausprägungen vom System akzeptiert werden.

Um sprechende Auswahloptionen auf der Oberfläche anzuzeigen, in der Datenbank aber nur die Literale speichern zu müssen, wird in Oracle APEX eine Werteliste angelegt. Die Werteliste in Abbildung 26 ermöglicht die Datenbankliterale auf der Benutzeroberfläche mit einem Anzeigetext einschließlich Übersetzungsvarianten. In der Benutzeroberfläche werden nun Auswahllisten genutzt, um den Anwender bei der Anwahl des Status zu unterstützen.

Sequenz	Anzeige	Zurück	Bedingt	Globalisierungstemplate	Aktualisiert
1	Bewerbung	B	-	-	15.05.19
2	Aktiv	A	-	-	15.05.19
3	Inaktiv	I	-	-	15.05.19

Abbildung 26: Anlage von Werteliste SCHUELER_STATUS in Oracle APEX (Quelle: Eigene Bildschirmkopie)

Zusätzlich besteht die Möglichkeit Wertelisten in Abhängigkeiten von Bedingungen einzuschränken. Sollte etwa bereits ein gewisser Status angewählt sein, könnten andere Status nicht mehr zur Verfügung stehen. Da jederzeit alle Status zur Auswahl stehen sollen, ist dies hier nicht erforderlich.

Es handelt sich hierbei, wenn auch sehr trivial, um die Abbildung einer prozessorientierten Geschäftslogik.

Änderungsdokumentation

Um eine Nachvollziehbarkeit von Änderungen zu gewährleisten, soll gemäß fachlicher Anforderung dokumentiert werden, zu welchem Zeitpunkt und durch welchen Benutzer ein Schüler erfasst und zuletzt bearbeitet wurde. Eine detaillierte Änderungsnachverfolgbarkeit ist nicht erforderlich. Da diese Anforderung nicht bereits automatisch durch die Low-Code Entwicklungsplattform berücksichtigt ist, muss diese individuell entwickelt werden.

Zur Umsetzung der Anforderung wird über die Oracle APEX Funktion SQL Workshop Object Browser zunächst die betroffene Tabelle *SCHUELER* um die erforderlichen Felder zur Speicherung dieser Daten erweitert.

Feldname	Datentyp	Beschreibung
SCHUELER.ERSTELLT	DATE	Zeitpunkt der Anlage
SCHUELER.ERSTELLT_VON	VARCHAR2(255)	Benutzername des Benutzers der Anlage durchgeführt hat
SCHUELER.AKTUALISIERT	DATE	Zeitpunkt der letzten Änderung
SCHUELER.AKTUALISIERT_VON	VARCHAR2(255)	Benutzername des Benutzers, der letzte Änderung vorgenommen hat

Tabelle 6: Erweiterung Tabelle SCHUELER für Änderungsdokumentation

Um automatisch bei Anlage oder Änderung von Schülerdaten diese Datenbankfelder zu befüllen, wird über die Funktion „Trigger erstellen“ in der Oracle APEX IDE ein Datenbanktrigger angelegt. Durch die Auswahl von *SCHUELER* als betroffene Tabelle mit den Ereignissen *INSERT* und *UPDATE* als Auslöser, wird ein Datenbanktrigger vorgeschlagen. Die Triggerfunktion muss mit PL/SQL Quelltext ergänzt werden, damit diese abhängig vom Ereignis die entsprechenden Datenfelder befüllt.

```
CREATE OR REPLACE TRIGGER SCHUELER_T1
BEFORE INSERT OR UPDATE ON SCHUELER for each row
BEGIN
    if inserting then
        :new.ERSTELLT := SYSDATE;
        :new.ERSTELLT_VON := sys_context('APEX$SESSION', 'APP_USER');
    else
        :new.AKTUALISIERT := SYSDATE;
        :new.AKTUALISIERT_von:= sys_context('APEX$SESSION', 'APP_USER');
    end if;
END;
```

Der PL/SQL Quelltext stellt über die Zuweisung „:new“ sicher, dass die Felder bei Anlage und Aktualisierung mit dem Systemdatum und dem Benutzernamen des in Oracle APEX authentifizierten Benutzers befüllt werden.

Eine Darstellung der Informationen ist nun auf der Benutzeroberfläche möglich. Hierfür muss lediglich die Betroffene Seite „Schüler“ in der Oracle APEX IDE über die Funktion Page Builder ausgewählt werden und per Rechtsklick auf „Spalten aktualisieren“ die neuen zusätzlichen Tabellenspalten berücksichtigt werden.

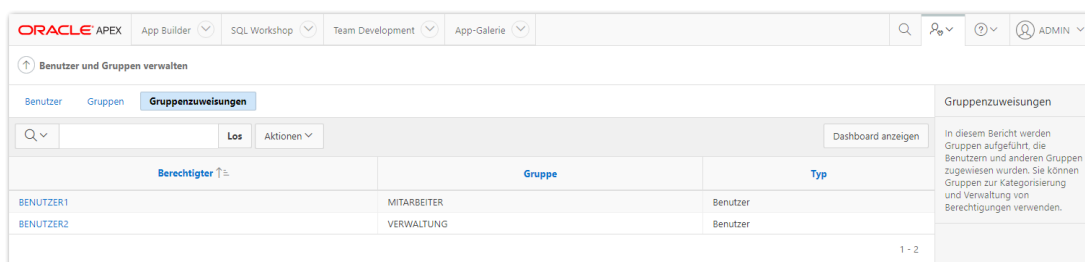
5.9 Beschränkung des Zugriffs

Die bisherige Entwicklung der Anwendung erfolgte ohne Berücksichtigung des Aspektes des eingeschränkten Zugriffs. Ein Zugriff wäre somit für jede Person möglich, die Kenntnis über die Webadresse der Geschäftsanwendung erhält und diese erreichen kann. Im folgenden Kapitel soll dies geändert werden und die Anforderung hinsichtlich Benutzerauthentifizierung und Autorisierung umgesetzt werden.

Basierend auf den fachlichen Anforderungen gilt es die nicht-authentifizierte Nutzung auszuschließen und authentifizierte Benutzer mit den Berechtigungsrollen Mitarbeiter und Verwaltung zu berücksichtigen.

Bei Anlage der Anwendung wurde durch Oracle APEX bereits das Authentifizierungsschema „Application Express Accounts“ angelegt. Es handelt sich hierbei um ein in Oracle APEX integriertes Benutzerverzeichnis, das ermöglicht Anwendungsbenutzer und Benutzergruppen über die Oracle APEX IDE zu erfassen. Das Verfahren hat gängige Funktionalitäten wie verschlüsselte Speicherung von Kennwortdaten, automatische Sperrung nach fehlerhaften Anmeldeversuchen und Passwort-Rücksetzungsfunktionalität bereits integriert.

In der Oracle APEX IDE werden über den Menüpunkt „Benutzer und Gruppen verwalten“ die erforderlichen Benutzergruppen *MITARBEITER* und *VERWALTUNG* angelegt. Je Benutzergruppe wird im Anschluss ein Benutzer angelegt, um im weiteren Verlauf die Nutzung der Geschäftsanwendung testen zu können.



Berechtigter	Gruppe	Typ
BENUTZER1	MITARBEITER	Benutzer
BENUTZER2	VERWALTUNG	Benutzer

Abbildung 27: Benutzer und Gruppen verwalten in Oracle APEX (Quelle: Eigene Bildschirmkopie)

Die Nutzung der Anwendung mit einem anonymen Benutzer ist nun nicht mehr möglich. Jeder Nutzer muss nun zunächst erfolgreich authentifiziert werden, um Zugriff

auf die Anwendung zu erhalten. Jedoch findet noch keine Beschränkung auf einzelne Anwendungsfunktionalitäten in Abhängigkeit der Benutzergruppe statt.

Es wurden folgende Benutzerdaten in der prototypischen Anwendung erfasst:

Benutzername	Kennwort	Benutzergruppe
BENUTZER1	Benutzer1#	MITARBEITER
BENUTZER2	Benutzer2#	VERWALTUNG

Tabelle 7: Benutzer und Gruppen der prototypischen Anwendung

Um die Funktionalitäten der Anwendung für einzelne Benutzer einzuschränken, ist die Anlage von Autorisierungsschema in Oracle APEX erforderlich. Ein Autorisierungsschema bündelt erforderliche Kriterien für die Autorisierung, wie in diesem Fall die Authentifizierung eines Benutzers einer speziellen Benutzergruppe. Das Autorisierungsschema prüft auf Basis des aktuell authentifizierten Benutzers ob die Bedingung zutrifft und gibt einen booleschen Wert zurück. Sollte der Rückgabewert des Schemas „wahr“ sein, ist die Autorisierung zutreffend, sollte der Wert „falsch“ sein, ist keine Autorisierung zutreffend.

Für die Anwendung werden deshalb folgende Autorisierungsschema in Oracle APEX angelegt:

Name	Beschreibung
IST_MITARBEITER	Benutzer wurde authentifiziert und gehört der Mitarbeitergruppe an.
IST_VERWALTUNG	Benutzer wurde authentifiziert und gehört der Verwaltungsgruppe an.

Tabelle 8: Autorisierungsschema der prototypischen Anwendung

Die definierten Autorisierungsschema können nun innerhalb der Anwendung eingesetzt werden, um Regionen, Funktionen und Seiten der Anwendung zu beschränken.

Autorisierungsschema

Abbrechen Löschen Änderungen anwenden

Alles anzeigen Name Subscription Autorisierungsschema Auswertungspunkt Kommentare

Name

Anwendung: 105 Ganztagsbetreuung

Name IST_VERWALTUNG

Subscription

Master-Autorisierungsschema referenzieren von Aktualisieren (Refresh)

Hierbei handelt es sich um die "Master"-Kopie dieses Autorisierungsschemas.

Keine Autorisierungsschemas abonnieren dieses Autorisierungsschema.

Autorisierungsschema

Schematyp Ist in Rolle oder Gruppe

Typ Workspace-Gruppe

Name(n) VERWALTUNG

Benutzer gehört nicht Gruppe Verwaltung an.

Fehlermeldung angeben, die bei einer Schemaverletzung aufgerufen wird

Mit Autorisierungsschemas können Sie Anwendungen, Seiten und Anwendungskomponenten schützen und den durch das Authentifizierungsschema der Anwendung bereitgestellten Schutz erweitern. Mit Autorisierungsschemas können Sie zusätzlichen Schutz über die einfache Benutzerauthentifizierung hinaus identifizieren. Beispiel: Ein Benutzer mit Administrationsberechtigungen muss möglicherweise auf mehr Navigationsleitsymbole, Seiten und Registerkarten zugreifen können als andere Benutzer.

Abbildung 28: Anlage eines Autorisierungsschema mit Oracle APEX IDE (Quelle: Eigene Bildschirmkopie)

Um zu vermeiden, dass die Datensätze der Einrichtungen und Gruppen von Benutzern außerhalb der Berechtigungsgruppe *VERWALTUNG* bearbeitet werden, wird dort jeweils im Abschnitt „Autorisierung bearbeiten“ das Autorisierungsschema *IST_VERWALTUNG* für die Operationen Hinzufügen, Aktualisieren und Löschen gemäß Abbildung 29 ausgewählt. Diese Operationen können folglich nur noch dann ausgeführt werden, wenn das Autorisierungsschema zutrifft, der authentifizierte Benutzer also der Benutzergruppe Verwaltung angehört.

Attribute

Filter

Bearbeiten

Aktiviert Ja Nein

Zulässige Vorgänge

Zelle hinzufügen

Zelle aktualisieren

Zelle löschen

Spalte "Zugelassene Zellenvorgänge"

- Auswählen -

Verlorenes Update

Zellenwerte

- Typ

Zelle hinzufügen, wenn leer.

Ja Nein

Autorisierung bearbeiten

Hinzufügen IST_VERWALTUNG

Aktualisieren IST_VERWALTUNG

Löschen IST_VERWALTUNG

Abbildung 29: Einschränkung der Bearbeitung auf Autorisierungsschema (Quelle: Eigene Bildschirmkopie)

5.10 Bereitstellung

Jegliche Änderung an der Geschäftsanwendung wird durch Bestätigung des Buttons „Speichern“ in der Oracle APEX IDE in der Low-Code Entwicklungsplattform gespeichert. Bei Aufruf der Anwendung durch einen Endnutzer erfolgt die Generierung von Oberfläche, Datenschicht und Geschäftslogik auf Basis dieser gespeicherten Konfigurationen. Eine Kompilierung oder manuelle Bereitstellung der Anwendung ist nicht erforderlich, da dies automatisiert durch die Low-Code Entwicklungsplattform erfolgt.

Die Geschäftsanwendung sowie jegliche Änderungen sind unmittelbar durch die Endanwender nutzbar. Hierfür muss lediglich im Webbrowser die Adresse *http://localhost:8080/f?p=105* aufgerufen werden. Der Aufrufparameter *p=105* entspricht der eindeutigen ID einer Oracle APEX Anwendung in einer Oracle APEX Umgebung.

In Tabelle 9 werden die unterschiedlichen Pfade die zur Entwicklung, Administration und Nutzung einer Oracle APEX Geschäftsanwendung erforderlich sind aufgelistet.

Anwendungspfad	Einsatzzweck	Beschreibung
/apex	Anwendungsentwicklung	Mandantenspezifische Oracle APEX IDE für Nutzung der Low-Code Entwicklungsplattform
/apex_admin	Instanzadministration	Mandantenübergreifende Administrationsoberfläche für administrative Verwaltung der gesamten Oracle APEX Instanz
/f?p=105	Anwendungsnutzung	Aufruf einer über Oracle APEX bereitgestellten Anwendung durch Endanwender, hier beispielhaft Anwendung mit eindeutiger ID 105.

Tabelle 9: Übersicht Anwendungspfad und Bestandteile

6 Auswertung

In den vorangegangenen Kapiteln wurden zunächst Methoden der Entwicklung mit geringem Anteil von Quelltext beschrieben, das Konzept von Low-Code Entwicklungsplattformen vorgestellt und anschließend eine Anwendung prototypisch auf Basis von Oracle APEX realisiert. Es soll nun ausgewertet werden, wie weit die Anforderungen an die Entwicklung einer Geschäftsanwendung durch diese Low-Code Entwicklungsplattform unterstützt und realisiert wurden.

6.1 Auswertungsmethodik

Im Rahmen der Auswertung soll zunächst ein Überblick über die durch die Low-Code Entwicklungsplattform erstellte Anwendung und deren Architektur geschaffen werden. Im Anschluss soll anhand des in Kapitel 4 entworfenen Anforderungskatalogs geprüft werden, in wie fern die Entwicklungsplattform erforderliche Anforderungen zur Entwicklung einer Geschäftsanwendung erfüllt. Für die Bewertung der Anforderungen werden folgende Bewertungskategorien herangezogen:

Bewertung	Beschreibung
+	Die Anforderung wird durch die Low-Code Entwicklungsplattform ausreichend erfüllt.
=	Die Anforderung wird durch die Low-Code Entwicklungsplattform nur in Teilen erfüllt oder es bestehen Einschränkungen, die jedoch, etwa durch Umgehungslösungen, erfüllt werden können.
-	Die Anforderung kann durch die Low-Code Entwicklungsplattform nicht erfüllt werden.

Tabelle 10: Beschreibung der Bewertungskategorien des Anforderungskatalogs

Abschließend wird geprüft, wieviel Quelltext tatsächlich für die Entwicklung der Geschäftsanwendung erforderlich war und die Plattform somit dem Low-Code Paradigma gerecht wurde. Dabei soll nicht nur die rein quantitative Menge von Quelltext sondern auch erforderliches IT Fachwissen und notwendige Programmierkenntnisse berücksichtigt werden.

6.2 Anwendungsarchitektur

Durch die Low-Code Entwicklungsplattform Oracle APEX wurde eine Geschäftsanwendung als Webanwendung generiert. Webanwendungen basieren auf dem DoD-Schichtenmodell. Hierbei wird die Anwendung über einen Webbrowser bereitgestellt

und der Anwender nutzt einen lokalen Webbrowser um auf die bereitgestellte Webseite zuzugreifen.⁵⁴ Andere Anwendungsmodelle wie z.B. native Applikationen werden durch Oracle APEX nicht unterstützt.

Der Vorteil bei dieser Art der Bereitstellung einer Anwendung liegt darin, dass auf den Computern oder mobilen Endgeräten der Nutzer keine Softwareinstallation erforderlich ist, die über den meist bereits vorinstallierten Webbrowser, hinausgeht. Als Nachteil dieser Art der Bereitstellung ist darauf hinzuweisen, dass für die Nutzung der Geschäftsanwendung stets eine Verbindung mit dem Webserver hergestellt sein muss. Die hier betrachtete Low-Code Entwicklungsplattform eignet sich somit weniger für Anwendungsfällen, bei denen die Geräte der Anwender keine oder nur eine instabile Verbindung zum Server aufbauen können. Für diese Anwendungsfälle wären Low-Code Entwicklungsplattformen besser geeignet, die eine offlinefähige Nutzung z.B. in Form einer nativen Betriebssystemanwendung oder Mobilgeräte Applikation ermöglichen.⁵⁵

6.3 Präsentationsschicht

Die Präsentationsschicht basiert im Rahmen der Standards für Webanwendungen auf der Auszeichnungssprache Hypertext Markup Language (HTML) in Kombination mit Cascading Stylesheets (CSS) für die Gestaltung.⁵⁶

Auffällig bei Betrachtung der Benutzeroberfläche sind interaktive Elemente wie z.B. die tabellarische Darstellung von Daten mit Sortierungs-, Gruppierungs- und Suchfunktionen. Um diese Funktionen für einen Benutzer ohne Neuladen der gesamten Webseite bereitzustellen, setzt Oracle APEX in der Kommunikation zwischen Webbrowser und Webserver auch auf den Einsatz von asynchronen Datenaufrufe.⁵⁷

Zur Gestaltung und Funktionalität der Oberfläche nutzt Oracle APEX Drittprojekte. Neben dem quelloffenen JavaScript Framework jQuery wird insbesondere das Oracle JavaScript Extension Toolkit (JET) eingesetzt. Es handelt sich hierbei um ein Ent-

⁵⁴ Vgl. Cerf & Cain, The DoD Internet Architecture Model, S. 309–311

⁵⁵ Vgl. Low-Code Plattformen für mobile Anwendungen in Kapitel 3.6

⁵⁶ Vgl. World Wide Web Consortium, Web Design and Application Standards

⁵⁷ Vgl. Mozilla Foundation, AJAX

wicklungsprojekt ebenfalls vom Hersteller Oracle Corporate, das auf Basis von JavaScript, HTML5 und CSS3 Technologien interaktive Komponenten für Benutzeroberflächen modular bereitstellt.

Die gesamte Benutzeroberfläche ist responsive gestaltet. Die Oberfläche ist dahingehend optimiert, dass sie auf unterschiedlichen Endgeräten mit unterschiedlichen Bildschirmgröße und Eingabeformen, z.B. Personalcomputer mit Bedienung via Maus und Smartphone mit Bedienung via berührungsempfindlichen Bildschirm, anwendbar ist. Eine Nutzung von gerätespezifischen Funktionalitäten, wie Sensoren für Positionsdaten eines mobilen Endgerätes oder spezifischer Oberflächenkomponenten z.B. Auswahlkomponente für Datum auf Smartphone, ist nicht vorgesehen.

Im Rahmen der prototypischen Realisierung hat sich herausgestellt, dass hiermit zwar die meisten Anforderungen auch per Smartphone nutzen lassen, die erstellte Anwendung aber hinsichtlich Oberflächendesign und Anwendungskomfort nicht mit denen einer spezifische für Smartphone entwickelten nativen Anwendung mithalten können. Sollte die Bedienung der Geschäftsanwendung primär durch mobile Endgeräte erfolgen, wäre sowohl aus Sicht der Oberflächengestaltung als auch offline-verfügbarkeit eine andere Low-Code Entwicklungsplattform zu evaluieren.

Es ist hervorzuheben, dass die gesamte Erstellung der Oberfläche automatisiert durch die Low-Code Entwicklungsplattform erfolgt ist und keinerlei Wissen zu den genannten Programmier- und Auszeichnungssprachen für die Weboberfläche bedurfte. Auch individuelle grafische Anpassungen wie z.B. die Farbgestaltung konnten über einen visuellen Editor leicht vorgenommen werden. Jedoch sind den Gestaltungsmöglichkeiten der gesamten Geschäftsanwendung bzw. einzelner Webseiten dadurch auch klare Grenzen gesetzt. Realisierbar ohne Quelltext ist nur, was im Rahmen des mitgelieferten Themes, vgl. dazu Abbildung 30, explizit als konfigurierbar definiert ist oder unter Auswahl der Standard Seitentemplates machbar ist.

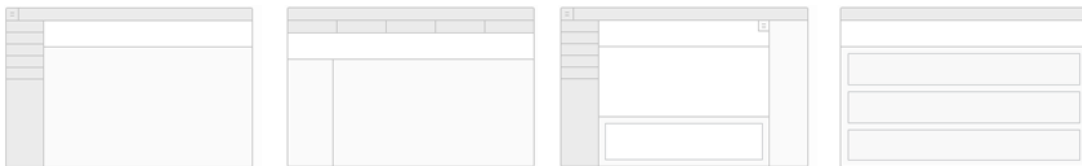


Abbildung 30: Standard Seitentemplates Oracle APEX (Quelle: Oracle Corporation)

Als besonders positiv ist die enge Verzahnung von Oberflächenkomponenten mit der Datenschicht anzumerken. Funktionalitäten wie z.B. Validierungen auf Ebene der Benutzeroberfläche oder die Bereitstellung von geeigneten Eingabefeldtypen wurden

durch die Low-Code Entwicklungsplattform anhand deren Datenstruktur im Datenbankmanagementsystem gewählt.

Bewertung der Anforderungen

Hinsichtlich der Präsentationsschicht kann folgende Bewertung der Anforderungen vorgenommen werden:

Nr.	Anforderung und Anmerkung	Wertung
1.1	Zeitgemäße Bereitstellungsart Einsatz von Webtechnologie für Benutzeroberfläche, jedoch keine Möglichkeit der alternativen Bereitstellung z.B. als Fat-Client oder native App	=
1.2	Adaptive Benutzeroberfläche Durch Einsatz responsiver CSS Technologien Anpassung der Oberfläche an unterschiedliche Endgeräte und Anwendungsszenarien	+
1.3	Positive Endnutzererfahrung Oberfläche entspricht aktuellen Standards, besonders durch Einsatz von Technologien wie AJAX zur Vermeidung von Neuladen von Webseiten, sehr positive Endnutzererfahrung	+
1.4	Präsentationslogik Low-Code Entwicklungsplattform unterstützt die Realisierung von Präsentationslogiken	+

Tabelle 11: Bewertung Anforderungen Präsentationsschicht

6.4 Datenschicht

Bei der im Prototyp eingesetzten Low-Code Entwicklungsplattform Oracle APEX nimmt das Datenbankmanagementsystem eine zentrale Rolle in der Plattform ein. Während bei anderen Low-Code Entwicklungsplattformen die zugrundeliegende Datenbanktechnologie für die Speicherung von Geschäftsdaten nicht bekannt und die Arbeit mit Daten stets über eine Abstraktionsschicht erfolgt, ist bei Oracle APEX der Typ der Datenbank bekannt und eine direkte Interaktion mit der Datenbank erforderlich. Eine Abstraktion der Datenschicht durch Methoden der modellgetriebenen Softwareentwicklung z.B. durch Beschreibung in UML wie in Kapitel 2.3 beschrieben, findet nicht statt. Aus Entwicklungssicht kann dies Vor- und Nachteile darstellen.

Als Vorteil kann gesehen werden, dass die Datenhaltung in Oracle APEX nicht innerhalb der eigentlich Low-Code Plattformkomponente erfolgt, sondern Anwendungsneutral in einer gängigen relationalen Datenbank. Dies ermöglicht die problemlose Nutzung der Anwendungsdaten jederzeit auch durch andere Geschäftsanwendungen. Wenngleich es sich bei der eingesetzten Datenbankmanagementsystem Oracle Database um eine herstellerspezifische Lösung handelt, ermöglicht diese über Standardschnittstellen jederzeit auch eine Anbindung anderer Anwendungstechnologien z.B. eine eigenentwickelte Java Geschäftsanwendung durch den SQL Java Database Connectivity (JDBC) Standard. Hersteller andere Low-Code Lösungen, die eine in die

Entwicklungsplattform voll eingebettete Persistenzschichten beinhalten, ermöglichen diesen direkten Zugriff auf die Daten nicht oder nur über spezifische Integrationschnittstellen. Es kann in diesen Fällen die Gefahr einer starken Abhängigkeit von einem Low-Code Hersteller entstehen, wenn die Daten nicht komfortabel durch andere Anwendungen weiterverwendet oder exportiert werden können (s.g. vendor-lock-in Effekt). Weiter schafft der Einsatz eines namhaften und im Umfeld von Unternehmen bereits etablierten Datenbankmanagementsystems Vertrauen und bietet eine Vielzahl von Optimierungs- und Nutzungsmöglichkeiten.

Durch die Möglichkeit der unmittelbaren Interaktion mit dem Datenbankmanagementsystem wird der Ansatz einer ganzheitlichen Low-Code Entwicklungsplattform möglicherweise auf gewichen. Es wird zwar in der Oracle APEX IDE eine Oberfläche zur assistierten Anlage und Verwaltung von Datenstrukturen bereitgestellt, zusätzlich bestehen aber durch den Zugriff auf die Datenbank via Datenbankadministrationswerkzeugen unterschiedlichste Hersteller zur Verfügung. Durch diese unterschiedlichen Werkzeuge können keine einheitlichen Konventionen, z.B. Groß-/Kleinschrift von Feldbezeichnungen, zur Realisierung der Datenstruktur sichergestellt werden und es besteht aufgrund der umfangreichen Funktionalitäten und Komplexität, z.B. Definition von Indexes und Constraints, die Gefahr der Überforderung von Anwendern, die kein ausreichendes Wissen zur Datenbankentwicklung mitbringen.

Alternative Werkzeuge zur Datenmodellierung

In der prototypischen Implementierung wurde der in Oracle APEX integrierte SQL Workshop Assistent genutzt, um die Datenstruktur zu modellieren. Der Assistent ermöglicht die Anlage von Datentabellen. Durchgeführte Schritte und gezielte Abfrage von relevanten Informationen einschließlich Hilfestellungen ist dies besonders für jemand ohne technischen Hintergrund einfacher. So wird z.B. anhand eines Feldnamens automatisiert durch APEX ein Feldtyp vorgeschlagen. Beispielsweise bei Feldern die den Bestandteil „Datum“ oder „Date“ haben automatisch der Feldtyp date. Die Anlage und Auswirkung von Indexes wird mit Hilfetexten erläutert und ist einfach durchführbar. Es sei darauf hingewiesen, dass über die SQL Workshop Oberfläche auch die Anlage von Ansichten, Prozeduren und weiteren Oracle SQL Datenbankobjekten möglich ist. Im Rahmen der prototypischen Anwendung wurden diese initial jedoch nicht benötigt.

Eine alternative zur reinen assistentenbasierten Anlage von Datenbankstrukturen stellt die Funktionalität Quick SQL in der Oracle APEX IDE dar. Im Sinne des Low-

Code Ansatzes können mit Quick SQL mit einer geringen Anzahl von Zeilen Datenbankstruktur angelegt werden.

Quick SQL ist eine simple und einfach verständliche Sprachsyntax zur Definition von Tabellenstrukturen, die durch Oracle APEX in valide SQL Statements transformiert und anschließend ausgeführt wird. Durch das Einrücken von Zeilen wird in Quick SQL eine Hierarchie abgebildet, die in einem relationalen Datenschema mündet. Sollte kein expliziter Datentyp vorgegeben werden, erfolgt die Wahl eines geeigneten Datentyps durch Quick SQL durch ein Regelwerk basierend auf dem Spaltennamen. Spalten die Beispielsweise im Namen „Datum“ oder „Date“ enthalten, werden als Datentyp Date angelegt.⁵⁸

Exemplarisch sollen in der Datenbank Schülerklassen (studentgroup) und in diesen Schüler (student) angelegt werden. Tabelle 12 zeigt die hierfür benötigt Quick SQL Syntax und die daraus generierten SQL Statements.

Quick SQL Syntax	Oracle SQL Output
<pre> studentgroup title student last_name first_name date_of_birth </pre>	<pre> create table studentgroup (id number not null constraint [...], title varchar2(4000)); create table student (id number not null constraint [...], studentgroup_id number constraint [...], last_name varchar2(255), first_name varchar2(255), date_of_birth date); create index student_i1 on student(studentgroup_id); </pre>

Tabelle 12: Vergleich Quick SQL und Oracle SQL zur Erstellung von Datenbanktabellen

Wenngleich das Beispiel aufzeigt, dass durch die Quick SQL Syntax mit wenig Zeilen Quelltext Datenbankstrukturen erstellt werden können, stößt diese hierarchisch ausgelegte Methode bei komplexeren relationalen Datenstrukturen, z.B. mehrfache Beziehungen zwischen Tabellen, schnell an ihre Grenzen.

⁵⁸ QuickSQL Referenzdokumentation: <https://docs.oracle.com/en/database/oracle/application-express/18.2/aeutl/using-quick-SQL.html>

Bewertung der Anforderungen

Hinsichtlich der Datenschicht kann folgende Bewertung der Anforderungen vorgenommen werden:

Nr.	Anforderung und Anmerkung	Wertung
2.1	Unterschiedliche Informationsarten Datenspeicherung erfolgt in relationalen Datenbankmanagementsystem, ermöglicht jedoch auch Speicherung von z.B. Dokumenten in binärformaten oder XML-Strukturen	+
2.2	Datenpersistenz Datenpersistenz erfolgt in relationalen Datenbankmanagementsystem unter Erfüllung der ACID Transaktionseigenschaften ⁵⁹	+
2.3	Datenmodellierung Datenmodellierung erfolgt unmittelbar in der relationalen Datenbank; es existiert keine vereinfachte Abstraktionsschicht. Gängige SQL Werkzeuge unterstützen visuelle Datenmodellierung.	=
2.4	Datenaustausch und Integration Unmittelbare Zugriffsmöglichkeiten auf Geschäftsdaten in Datenbank; es bestehen weitreichende Möglichkeiten für Datenaustausch und Integration	+

Tabelle 13: Bewertung Anforderungen Datenschicht

6.5 Geschäftslogik

Der Fokus hinsichtlich Anwendungslogik von Oracle APEX liegt auf den s.g. CRUD-Funktionalitäten, darunter werden die Aktionen zur Anlage, zum Lesen, zum Ändern und zum Löschen von Daten verstanden.

Interaktion innerhalb Anwendung

Die Interaktion innerhalb der Benutzeroberfläche erfolgt durch Hyperlinks in der Webseitenstruktur. Das Springen von einer Seite zu einer anderen Seite kann durch den Benutzer jederzeit durch einen Klick ausgelöst werden. Es besteht keine Möglichkeit schrittweise bzw. prozessuale Interaktionen auf Eben der Oberfläche durch die Anwendungslogik zu realisieren, wie es z.B. für eine Dialogassistenzmaske erforderlich wäre.

Eingabevalidierung

Oracle APEX sieht die Eingabevalidierung auf unterschiedlichen Schichten vor:

⁵⁹ ACID, in Deutsch auch AKID Eigenschaften: Atomarität, Konsistenzerhaltung, Isolation, Dauerhaftigkeit

- **Datenbankschicht:** Die unterste mögliche Schicht der Datenvalidierung stellt die Datenbank dar. Durch s.g. Constraints können Bedingungen bei Einfügen, Änderung und Löschen von Datensätzen in der Datenbank definiert werden.
- **Anwendungsschicht:** Die Oracle APEX Anwendungsschicht ermöglicht die Definition unterschiedlicher Validierungen auf Feld- und Zeilenebene. Die Art der Validierung kann entweder aus vordefinierten Kriterien (z.B. Eingabe bestehend aus Anzahl von Zeichen) ausgewählt oder individuell durch PL/SQL Quelltext entwickelt werden. Die individuelle Entwicklung unter Einsatz von PL/SQL ermöglicht auch die Abfrage von anderen Datensätzen, um z.B. auch fachliche Dubletten prüfen zu können.
- **Oberflächenschicht:** Auf der Schicht der Oberfläche sind durch Nutzung der Scriptsprache JavaScript Eingabevalidierungen möglich. Hierbei gilt es stets zu berücksichtigen, dass bei Webanwendungen die Oberfläche im Browser des Endanwenders präsentiert wird und Manipulationen auf dieser Ebene nicht ausgeschlossen werden können. Einem Endanwender wäre es somit möglich, diese Ebene der Validierung zu umgehen. Oracle APEX sieht bei der Wahl von Validierungen auf Ebene der Anwendungsschicht zusätzliche Validierungen auf der Oberflächenschicht vor, die aber nur zusätzlich wirken um den Nutzer mögliche Fehler bereits vor Absenden der Dateneingabe zu präsentieren.

Die im Prototyp exemplarisch implementierte Validierung der Eingabedaten erfolgt auf Ebene der Anwendungsschicht und nicht innerhalb der Datenpersistenzschicht. Die Validierung erfolgt ohne Sperrung der betroffenen Tabelle der Datenbank. Ohne Sperrung kann nicht ausgeschlossen werden, dass im Zeitfenster zwischen erfolgter Validierung und dem tatsächlichen Einfügen der Zeile in die Datenbank ein paralleler Schreibvorgang erfolgt ist. Aufgrund der anzunehmenden geringen Anzahl von zeitgleichen Schreibvorgängen und des kurzen Zeitfensters wird dennoch diese Lösung gewählt. Alternativ wäre eine Realisierung über einen Datenbanktrigger innerhalb der Persistenzschicht einschließlich der Sperrung (Locking) der betroffenen Datensätze möglich. Diese Funktionalität wird aber nicht durch Oracle APEX auf Anwendungsschichtebene bereitgestellt.⁶⁰

⁶⁰ Vgl. Kemp, Non-Overlapping Dates Constraint

Geschäftsprozesse

Oracle APEX stellt keine Funktionalität zur Abbildung von Prozessabläufen bereit. Im Unterschied zu unmittelbar vergleichbaren Low-Code Entwicklungsprodukten, wie z.B. OutSystems oder Mendix, ist die Abbildung von Geschäftsabläufen nicht deklarativ in Form von geeigneten Modellen oder Sprachen möglich möglich.⁶¹ Um dennoch an das gewünschte Ziel zu gelangen bestehen nach Bruijn folgende Möglichkeit zur Realisierung von einfachen und komplexen Geschäftsprozessen in Oracle APEX Anwendungen:⁶²

- **Programmierung PL/SQL:** Die Anforderung kann über die Kombination eines Status-Feldes in der Datenbank mit entsprechender Geschäftslogik in der Programmiersprache PL/SQL umgesetzt werden. Über das Statusfeld kann der aktuelle Zustand im Prozess dokumentiert werden. Abhängig von der Komplexität des Prozesses kann die Kombination aus Programmquelltext und Datenbankstatusfeld schnell eine hohe Komplexität erreichen und schlecht wartbar werden. Kritisch ist hier auch die Vermischung von Geschäftsdaten und Statusinformationen in einer Datentabelle zu sehen.
- **Oracle APEX Prozessplugins:** Die funktionelle Lücke aber auch die gute Erweiterbarkeit von Oracle APEX hat dazu geführt, dass zwei unabhängige Erweiterungsprojekte entstanden sind, um den Mangel an Prozesslösungen auszugleichen. Das Projekt Flow Control Toolkit (FCT) ermöglicht es in separaten Datenbanktabellen Zustände, Zustandsübergänge und Aktionen in Anlehnung an das Konzept eines Endlichen Automaten zu definieren und diese Funktionalität gekapselt über PL/SQL Packages aufzurufen. Im Unterschied dazu setzt das Projekt PL/FLOW Geschäftsprozesse nach dem Workflow Management Coalition Workflow Standard, ebenfalls in separaten Datenbanktabellen und Zugriff via PL/SQL Paketen um.
- **Externe Workflow-Management-Systeme:** Eine Möglichkeit stellt die Anbindung eines externen Workflow-Management-Systems, wie die OpenSource Lösung Camunda dar. Dieses System basiert auf der BPMN und ermöglicht somit eine deklarative und standardisierte Beschreibung von Geschäftsprozessen. Die Fachlichkeit der Prozesse und deren Ablaufsteuerung erfolgt hier komplett

⁶¹ Vgl. Mendix Microflows (<https://docs.mendix.com/refguide/microflows>) und OutSystems Business Process Technology (https://success.outsystems.com/Documentation/Best_Practices/Notation_Reference_for_BPMN_Practitioners)

⁶² Bruijn, Affordable Workflow Options for your APEX App

außerhalb von Oracle APEX. Durch APEX erfolgt nur bei Ereignissen ein Anstoß an oder Benachrichtigung durch das Workflow-Management-System. Da beide Systeme das REST Protokoll unterstützen, ist die Integration der Systeme möglich.

Da die im Rahmen der prototypisch zu realisierender Anwendung die Anforderung an Geschäftsprozesse relativ trivial waren, wurden diese lediglich durch ein Feld zur Speicherung des Status eines Schülers im Datenbankmanagementsystem umgesetzt.

Bewertung der Anforderungen

Hinsichtlich der Geschäftslogik kann folgende Bewertung der Anforderungen vorgenommen werden:

Nr.	Anforderung und Anmerkung	Wertung
3.1	Kalkulation und Aggregation Kalkulation und Aggregation von Daten nur auf Ebene der Datenschicht durch geeignete SQL und PL/SQL Befehle, keine Logikschicht	=
3.2	Lesen und Modifikation von Daten Sehr einfache und leistungsfähige Bereitstellung von CRUD Funktionalitäten für Lesen und Modifikation von Daten	+
3.3	Datenvalidierung und Plausibilisierung Validierung und Plausibilisierung erfolgt auf unterschiedlichen Schichten, einfache und unterstützte Einrichtung möglich	+
3.4	Geschäftsprozesse Abbildung komplexer Geschäftsprozesse nicht vorgesehen, Umgehungs-lösungen z.B. durch Anbindung externer Systeme möglich	=
3.5	Integration von externer Geschäftslogik Anbindung externer Geschäftslogik z.B. durch Integrationsschicht nicht vorgesehen, programmatischer Aufruf von Webservices in Datenschicht jedoch möglich	=

Tabelle 14: Bewertung Anforderungen Geschäftslogik

6.6 Querschnitts- und Unterstützungsfunktionen

Die Vorteile der Low-Code Entwicklungsplattform wurden insbesondere bei der Berücksichtigung von Querschnittsfunktionen deutlich. Durch einen stets ähnlichen Bedarf an Funktionalitäten in Geschäftsanwendungen, stehen diese Funktionen in der Plattform zentral bereit und müssen nicht individuell entwickelt werden. Die Funktionen müssen lediglich konfiguriert und im fachlichen Anwendungskontext eingesetzt werden. Beispielsweise wurde im Prototyp die Funktionalität der Benutzerauthentifizierung und Autorisierung durch die Low-Code Plattform bereitgestellt. Durch den Entwickler musste nur die Art der Authentifizierung sowie des Benutzerverzeichnisses konfiguriert und die zu schützenden fachlichen Daten markiert werden.

Benutzerauthentifizierung und Autorisierung

Statt die Funktionalitäten zur Authentifizierung und Autorisierung individuell zu entwickeln, stellt die Oracle APEX Low-Code Entwicklungsplattform Module bereit, die lediglich an die individuellen Bedürfnisse angepasst und konfiguriert werden müssen. In der prototypischen Anwendung werden zur Authentifizierung die integrierten Benutzeraccounts von Oracle APEX verwendet. Gängige Anforderungen an das Benutzermanagement wie z.B. starke Passwortverschlüsselung, Passwort-Rücksetzung und Schutz vor Angriffsmethoden der rohen Gewalt (engl. brute-force) sind bereits implementiert. Eine individuelle Entwicklung entfällt.⁶³

Alternativ stellt die Low-Code Entwicklungsplattform aber auch Authentifizierungsstandards wie SAML, OAuth2 oder LDAP Benutzerverzeichnisse vorkonfiguriert bereit. Durch Nutzung dieser Standards ist eine Integration der Geschäftsanwendung in bestehende Unternehmenssysteme z.B. durch Nutzung von Single-Sign-On Verfahren möglich.

Von der Benutzerauthentifizierung gekapselt werden durch die Low-Code Entwicklungsplattform Methoden zur Autorisierung bereitgestellt. Hiermit lassen sich unabhängig vom eingesetzten Authentifizierungsverfahren Beschränkungen auf Seitenkomponenten realisieren.⁶⁴

Anwendungs- und Datensicherheit

Das Open Web Application Security Project (OWASP) beschreibt die Top 10 Risiken für die Anwendungssicherheit von Webanwendungen.⁶⁵ Die zu realisierende Anwendung soll ein angemessenes Sicherheitsniveau erreichen und gegen die bekannten Angriffsvektoren gesichert sein. Exemplarisch soll hier ausgewertet werden, ob die drei gängigsten Risiken und deren Mitigation durch die Low-Code Entwicklungsplattform unterstützt wurden.

Injection-Schwachstellen treten auf, wenn nicht vertrauenswürdige Daten als Teil einer Datenbankabfrage interpretiert werden. Es besteht das Risiko, dass der Angreifer den Teil der Abfrage so modifiziert, dass er unautorisiert auf Daten oder Funktionen zugreifen kann. Tatsächlich werden an einer Vielzahl von Stellen in der Anwendung SQL Abfragen basierend auf dynamischen, nichtvertrauenswürdigen Daten, erstellt.

⁶³ Vgl. Spendolini, Expert Oracle Application Express Security, S. 159–160

⁶⁴ Vgl. Spendolini, Expert Oracle Application Express Security, S. 177

⁶⁵ Vgl. Open Web Application Security Project, OWASP Top 10 Risiken für die Anwendungssicherheit

Die Seite zur Darstellung von Schülerdetails nutzt folgende Abfrage in Abhängigkeit der über die URL-Zeile übergebene eindeutiger ID eines Schülers:

```
select * from SCHUELER where id = :P1_ID
```

Sollte keine Absicherung durch die Anwendung bzw. durch den SQL Interpreter erfolgen, würde dies ein Risiko für einen SQL Injection Angriff darstellen. Die Übergabe der dynamischen Information erfolgt hier und standardgemäß in Oracle APEX als Bind-Variable in der Doppelpunktnotation. Bind-Variablen werden erst nach dem einlesen und analysieren der SQL-Abfrage ersetzt. Dies bedeutet, dass der Inhalt der Variable technisch keine Auswirkung auf die Struktur der Abfrage haben kann. Ein SQL Injection Angriff ist somit bei Nutzung der Bind-Variablen ausgeschlossen.⁶⁶ Eine Ausnahme können dynamisch generierte SQL-Anweisungen darstellen, die zur Laufzeit aus Zeichenketten konkatiert und dynamisch ausgeführt werden. Von dieser Art von dynamisch generierten SQL-Anweisungen wird explizit abgeraten.⁶⁷

Ein weiteres Risiko nach OWASP besteht für die Anwendungssicherheit bei Anwendungsfunktionen, die die Authentifizierung und das Session-Management fehlerhaft implementieren. Dies kann Angreifern ermöglichen die Benutzersessions fremder Personen zu übernehmen. Gemäß vorangegangener Beschreibung werden die Mechanismen zur Authentifizierung und zum Session-Management von Oracle APEX verwendet. Die Anwendung setzt somit auf diesen Standard-Mechanismen des Frameworks auf und implementiert diese nicht selbst. Durch die langjährige und professionelle Entwicklung von Oracle APEX ist von einem hohen Sicherheitsstandard dieser Komponenten auszugehen.

Als Cross-Site Scripting werden Schwachstellen verstanden, bei denen nichtvertrauenswürdige Daten an einen Webbrowser gesendet werden. Ein Angreifer kann diese Schwachstelle ausnutzen, um mit schadhaftem Quelltext die Webseite bei Aufruf durch andere Benutzer zu verändern oder an Seiteninhalte zu gelangen. Spalten in Reports und Seitenelemente werden standardgemäß mit der aktiven Eigenschaft „Escape bei Sonderzeichen“ im Oracle APEX Pagedesigner angelegt. Sollten potenziell schadhafte Steuerungszeichen ausgegeben werden, werden diese Steuerungszeichen vor Ausgabe bereinigt.⁶⁸

⁶⁶ Vgl. Spendolini, Expert Oracle Application Express Security, S. 129–131

⁶⁷ Vgl. Sieben, Oracle APEX, S. 814

⁶⁸ Vgl. Sieben, Oracle APEX, S. 815–818

Eine weitergehende Analyse der OWASP Top 10 ist in der Oracle Express Community vorzufinden.⁶⁹ Zur langfristigen Wahrung eines angemessenen Sicherheitsstandards sollten halbjährlich oder jährlich erscheinenden Updates der Low-Code Plattform Oracle APEX verwendet werden, um von Fehlerbehebungen und verbesserten Sicherheitsmechanismen profitieren zu können. Grundsätzlich ist bei der Nutzung Plattform aber bereits von einem guten und qualitativ hochwertigen Standard hinsichtlich Anwendungssicherheit auszugehen. Laut des zentralen Common Vulnerabilities and Exposures (CVE) Verzeichnisses sind seit 2008 nur fünf Sicherheitslücken in Application Express bekannt.⁷⁰

Zusammenarbeit und Kollaboration

Die Arbeit am Programmquelltext erfolgt in klassischen Modellen der Anwendungsentwicklung meist durch einen Benutzer auf einem lokalen Computer. Änderungen müssen im Anschluss an andere Entwickler verteilt werden. Dies geschieht meist durch die Bereitstellung des geänderten Programmcodes in einer Repository. Andere Entwickler können dann auf den geänderten Quelltext zugreifen und diesen in ihren eigenen Änderungen integrieren.

Durch den webbasierten Ansatz der Low-Code Entwicklungsplattform Oracle APEX existiert kein lokaler Quelltext, sondern jegliche Änderung, unabhängig ob visuell über Oberfläche oder in Form von z.B. PL/SQL programmiert, ist sofort für alle Entwickler transparent. Ein Zeitverzug durch Austausch von Quelltext entfällt. Jedoch kann dies auch ein Risiko darstellen, wenn z.B. ohne gegenseitiges Wissen Entwickler gleichzeitig dieselben Komponenten bearbeiten. Oracle APEX bietet die Funktion der exklusiven Sperre von Anwendungskomponenten. Durch das Setzen von Sperren wird im System dokumentiert, welcher Entwickler aktuell an welchem Teilbereich arbeitet und verhindert effektiv das gleichzeitige Bearbeiten eines Teils der Anwendung.⁷¹

Anwendungsdebugging

Zur Diagnostizierung und Behebung von Fehlersituationen kann es erforderlich sein, die Anwendung oder Teile der Anwendung mit einem geeigneten Werkzeug zu ana-

⁶⁹ Vgl. Möckel, Oracle Application Express unter der Security-Lupe

⁷⁰ Im Vergleich dazu sind zu dem PHP basierten Web-Contentmanagementsystem WordPress aktuell 286 Sicherheitslücken bekannt geworden: <https://www.cvedetails.com/>, beide Anwendungen ohne Betrachtung von Datenbank oder Webserver

⁷¹ Vgl. Peake, Life Cycle Management with Oracle Application Express, S. 11

lysieren. Bei der klassischen Programmierung wird hierfür meist ein Debugger genutzt, der die Steuerung des Programmablaufs, etwa durch Definition von Haltepunkten im Quelltext und Einzelschritt-Verarbeitung von Befehlen, ermöglicht. Durch den Plattformansatz, der teils keinen Einblick in den ausgeführten Quelltext ermöglicht, ist kein Debugging in diesem herkömmlichen Sinne für die Anwendung möglich.

Um Systemzusammenhänge und Fehlersituationen analysieren zu können, besteht die Möglichkeit eine detaillierte Protokollierung von Programmabläufen zeitweise zu aktivieren. Dadurch werden verwendete Parameter und Methodenaufrufe im System gespeichert und können nachvollzogen werden.⁷²

Für individuell Entwickelte Erweiterungen der Kernfunktionalität von Oracle APEX besteht die Möglichkeit über externe Werkzeuge ein Debugging durchzuführen. Dies berücksichtigt aber stets nur die individuell entwickelten Komponenten und nicht ganzheitlich Programmabläufe in der Low-Code Entwicklungsplattform. So kann beispielsweise in der Entwicklungssoftware Oracle SQL Developer Quelltext der Sprache PL/SQL schrittweise ausgeführt werden.

Anforderungs- und Projektmanagement

In der Oracle APEX IDE werden im Anwendungsbereich „Team Development“ Funktionalitäten zur Dokumentation von Anforderungen, Meilensteinen, Fehlern und Fortschrittsberichten bereitgestellt. Es handelt sich um umfangreiche Funktionalitäten, die insbesondere durch die Möglichkeit der Aufgabenzuweisung an einzelne Entwickler und der direkten Integration in die APEX Oberfläche hilfreich sind. Durch die direkte Integration können innerhalb der Anforderungs- und Projektmanagement Werkzeuge auch direkte Beziehungen zu einzelnen Funktionen der Anwendung hergestellt werden, z.B. die Verknüpfung der betroffenen Datentabellen oder Anwendungsseiten.⁷³

Sollten diese Funktionalitäten nicht bestehen, wäre es erforderlich durch externe Software diese Steuerung des Anforderungs- und Projektmanagements zu übernehmen.

Anwendungstest

Durch die Vielzahl an bereits mitgelieferten Funktionalitäten der Low-Code Entwicklungsplattform ist der sonst übliche Testansatz, der möglichst alle Schichten und Funktionalitäten der Anwendungssoftware berücksichtigt, nicht erforderlich und auch

⁷² Vgl. Sieben, Oracle APEX, S. 683

⁷³ Vgl. Sieben, Oracle APEX, S. 111–112

nicht sinnvoll. Grund hierfür ist, dass davon auszugehen ist, dass vom Hersteller bereitgestellte Funktionalitäten bereits vorab getestet und in der Annahme fehlerfrei bereitgestellt werden. Dies grenzt den Bedarf an erforderlichen Testfällen ein, da nur noch selbstentwickelte Funktionalitäten der Geschäftsanwendung berücksichtigt werden müssen. Dies betrifft insbesondere individuelle Geschäftslogiken die mit der Programmiersprache PL/SQL erstellt wurden sowie Anpassungen an den Benutzeroberflächen in den Auszeichnungssprachen HTML und CSS.

Die Low-Code Entwicklungsplattform Oracle APEX bietet keine Funktionalitäten zur Erstellung und Ausführung von Testfällen. Die Verwendung von gängigen Werkzeugen, im Prototyp Oracle SQL Developer für Modultests von PL/SQL Quelltext und Selenium für Weboberfläche, ermöglichen jedoch die Sicherstellung von Testfällen. Um Besonderheiten der dynamisch generierten Weboberfläche in automatisierten Tests zu berücksichtigen, existiert mit APEX Test Automation Framework (ATAF) ein Framework um vereinfacht auf einzelne Komponenten einer Webseite zugreifen zu können.⁷⁴

Es wurde im Rahmen der Anwendung individueller PL/SQL Quelltext erstellt, der eine Testabdeckung zum Ausschluss von gängigen Fehlerkonstellationen erfordert. Die Oracle APEX IDE stellt keine Funktionalität zum Testen von eingesetzten PL/SQL Quelltext bereit, jedoch ist ein Modultest über die Datenbankadministrationswerkzeuge, hier Oracle SQL Developer, einfach möglich. Über die Funktion Extras / Einheitentest werden Testfälle erstellt, die gemeinsam mit dem Quelltext der PL/SQL Funktionen in der Datenbank abgelegt werden. In Abbildung 31 wird die Ausführung eines Testfalls für die individuell entwickelte PL/SQL Funktion VALID_RANGE aufgezeigt.

Testlauf	Status	Dauer	Meldung
VALID_RANGE			
✓ PSTHKOELN: Ausführen am - 2019-05-28 12:54:17.8855	SUCCESS	10	
✓ Implementierung - Testimplementierung 1	SUCCESS	10	
✓ Vorgangsaufruf	SUCCESS	10	
✓ <RETURN>	SUCCESS		Erwartet: [1], Empfangen: [1]
IN-Parameter #1 - VALID_FROM			Wert: [01.05.19]
IN-Parameter #2 - VALID_UNTIL			Wert: [10.05.19]
IN-Parameter #3 - COMPARE_DATE			Wert: [08.05.19]

Abbildung 31: Testausführung mit Oracle SQL Developer (Quelle: Eigene Bildschirmkopie)

⁷⁴ Produktwebseite ATAF <https://github.com/schunt1/ATAF>

Neben dem Oracle SQL Developer gibt es noch weitergehende und spezialisierte Anwendungen für die Ausführung von Modul- und Integrationstests von PL/SQL Quelltext.⁷⁵

Da es sich bei der durch Oracle APEX generierten Oberfläche um eine HTML Webseite handelt, kann zusätzlich zum Oberflächentest das quelloffene Test-Framework Selenium eingesetzt. Selenium ist ein Werkzeug, dass die automatisierte Bedienung einer Weboberfläche simuliert und prüft, ob die Oberfläche der erwarteten Testkriterien entspricht.

Änderungsnachverfolgbarkeit

Die Nachvollziehbarkeit aber auch Rücksetzung von Quelltextänderungen ist in Entwicklerteams nicht nur bei versehentlichen Änderungen ein sehr nützliches Hilfsmittel, sondern bei umfangreichen Geschäftsanwendungen oft auch eine Pflicht, um Missbrauch der Anwendung auszuschließen. Gängige Anwendungen für die Quelltextversionierung sind Git, Subversion oder SVN.

Da in Oracle APEX kein Quelltext in diesem Sinne greifbar ist, können diese Werkzeuge nicht eingesetzt werden. Auf Anwendungsebene wird in Oracle APEX eine Änderungshistorie fortgeschrieben, die aber nicht granular die Änderungen enthält und nicht rückgängig gemacht werden kann. Da sämtliche Änderungen in die Datenbank geschrieben werden, besteht die Möglichkeit auf Ebene der Datenbank über Backups und der Funktionalität von Datenbank Änderungsprotokollen (Redo- und ArchiveLogs) Änderungen nachzuvollziehen. Dies stellt aber keine wirklich komfortable Lösung dar.

Umgebungstrennung

Durch den webbasierten Entwicklungsansatz von Oracle APEX und Nutzung des Datenbankmanagementsystems zur Verwaltung des Anwendungsquelltext, Metadaten und Definitionen, entfallen sonst üblichen Zusatzanforderungen wie z.B. Versionsverwaltungssoftware. Weiter sind Hilfsmittel zur Projekt- und Arbeitsorganisation unmittelbar in der Entwicklungsplattform berücksichtigt, was zusätzliche Anwendungen erspart.

Die Low-Code Entwicklungsplattform sieht grundsätzlich nur eine Systemumgebung vor, in der vorgenommene Änderungen unmittelbar bereitstehen. Eine Umgebungs-

⁷⁵ Vgl. Sieben, Oracle APEX, S. 699–700

trennung nach z.B. Entwicklungs-, Test- und Produktionssystem ist somit nicht gegeben. Dies kann insbesondere für komplexere Geschäftsanwendungen ein Risiko darstellen, da vorgenommene Änderungen einschließlich möglicher Entwicklungsfehler, unmittelbar die produktive Geschäftsanwendung betreffen.

Der Hersteller zeigt jedoch in einem Whitepaper Lösungsansätze zu Entwicklung, Test und produktiver Bereitstellung von Anwendungen in getrennten Systemumgebungen auf.⁷⁶ Auch Sieben empfiehlt für die Entwicklung von komplexeren APEX-Anwendungen den Aufbau einer Deployment-Strategie die aus mehreren Systemumgebungen bestehen kann und Konfigurationszustände per Import- und Exportfunktionalitäten austauschen.⁷⁷

Anwendungsbereitstellung

Es werden unterschiedliche Nutzungs- bzw. Bereitstellungsoptionen der Low-Code Entwicklungsplattform angeboten. Es ist explizit auch ein Wechsel zwischen den Bereitstellungsoptionen möglich.

- **Lokale Bereitstellung:** Die Installation der Low-Code Entwicklungsplattform bzw. der hierfür erforderlichen Komponenten ist auf einem lokalen Computer möglich. Die erforderlichen Komponenten (insbesondere Oracle Database) werden als Installationsmedien für Windows und Linux-basierte Betriebssysteme bereitgestellt. Für den Einsatz auf einem lokalen Computer wird die Variante Oracle Database Express Edition (XE) empfohlen, da diese lizenzkostenfrei eingesetzt werden kann und weniger Ressourcen erfordert. Diese Bereitstellungsvariante ist insbesondere für die schnelle und einfache Softwareentwicklung ohne Server und ohne Bedarf einer Internetverbindung geeignet. Aufgrund der erforderlichen Ressourcen ist ein produktivbetrieb unter Nutzung eines Laptops als Anwendungsservers eher nicht sinnvoll.
- **On-Premises Bereitstellung:** Als On-Premises Bereitstellung wird der Einsatz der Low-Code Plattform auf einer lokalen Serverinfrastruktur des einsetzenden Unternehmens verstanden. Bis ins Jahr 2010 galt diese Variante als der Normalfall bei Betrieb von serverbasierten Anwendungen, erst durch die zunehmende Verbreitung von cloud-basierten Lösungen hat sich überhaupt diese explizite Bezeichnung entwickelt. Für den Einsatz in Serverumgebungen eigenen

⁷⁶ Vgl. Peake, Life Cycle Management with Oracle Application Express, S. 4

⁷⁷ Vgl. Sieben, Oracle APEX, S. 723–728

sich laut Hersteller insbesondere die für Mehrkernprozessoren und hohen Hauptspeicher ausgelegten Enterprise Versionen von Oracle Database, jedoch ist auch hier ein Betrieb von Oracle XE möglich.

- **Cloud-basierte Bereitstellung:** Als cloud-basierte Bereitstellung wird verstanden, dass die Low-Code Plattform vollständig durch einen externen IT-Dienstleister betrieben und von einem Unternehmen als Dienstleistung eingesetzt wird. Das nutzende Unternehmen benötigt selbst keinerlei Infrastruktur. Als Vorteil dieser Variante wird insbesondere die sehr schnelle Einsatzfähigkeit der Plattform hervorgehoben, da die Nutzung bei Cloud-Anbietern wie z.B. Oracle selbst oder Amazon Web Services (AWS) bereits nach wenigen Minuten möglich ist.

Unabhängig vom gewählten Bereitstellungsmodell ist Oracle APEX mandantenfähig ausgelegt. Dies bedeutet, dass über eine Instanz der Low-Code Plattform unterschiedliche Nutzergruppen, bei Bedarf auch logisch und physikalisch in der Datenbank getrennt, bedient werden können. Auf dieser Eigenschaft setzen die Cloud-Anbieter auf und stellen dem Nutzer innerhalb einer Oracle APEX Instanz einen Nutzungsbereich (Workspace) bereit.⁷⁸

Bewertung der Anforderungen

Hinsichtlich der Querschnitts- und Unterstützungsfunktionen kann folgende Bewertung der Anforderungen vorgenommen werden:

Nr.	Anforderung und Anmerkung	Wertung
4.1	Anwendungs- und Datensicherheit Gängige Risiken nach OWASP werden berücksichtigt	+
4.2	Benutzerauthentifizierung und Autorisierung Bereitstellung unterschiedlicher Authentifizierungs- und Autorisierungsverfahren	+
4.3	Kollaborative Anwendungsentwicklung Webbasierte Anwendungsentwicklung ermöglicht kollaborative Vorgehensweise	+
4.4	Änderungsnachverfolgbarkeit Kein Einsatz gängiger Werkzeuge zur Änderungsnachverfolgbarkeit möglich, Nachvollziehbarkeit über Protokollierung in Datenbank möglich	=
4.5	Anwendungsdebugging Debugging der Applikation wird über integrierte Funktionen der Low-Code Entwicklungsplattform unterstützt	+
4.6	Anwendungstest Klassischer Testansatz nicht erforderlich, jedoch Bereitstellung geeigneter Hilfsmittel für automatisierte Oberflächentests	+

⁷⁸ Vgl. Sieben, Oracle APEX, S. 723

4.7	Anwendungsbereitstellung Es existieren unterschiedliche Bereitstellungsmodelle, sowohl cloud-basiert als auch auf eigener Infrastruktur	+
4.8	Umgebungstrennung Grundsätzlich als Einzelumgebung vorgesehen, jedoch Lösungsansätze vorhanden, um Mehrumgebungsszenarien zu realisieren	=
4.9	Ereignisprotokollierung Integrierte Funktionalitäten zur Protokollierung von Ereignissen	+

Tabelle 15: Bewertung Anforderungen Querschnitts- und Unterstützungsfunktionen

6.7 Anwendung des Low-Code Paradigmas

Ein Kernversprechen von Low-Code Entwicklungsplattformen ist der geringe Bedarf an Quelltext zur Realisierung der Geschäftsanwendung. Es soll ausgewertet werden, wieviel Quelltext zur Realisierung der prototypischen Anwendung erforderlich war.

Zur Realisierung wurden gängige Standards und Konventionen der Low-Code Entwicklungsplattform Oracle APEX eingehalten.⁷⁹ Das spiegelt sich beispielsweise an der Oberflächengestaltung wider, die auf das integrierte Oracle APEX Universal Theme aufsetzt und dadurch kaum Anpassungen erforderte. Die Positionierung sämtlicher Inhalts- und Gestaltungselemente erfolgte über die Gestaltungsoberfläche Oracle APEX Page Builder ohne Quelltext.

In Tabelle 16 wird die Menge des verwendeten Quelltextes dargestellt. Die Berechnung der Quelltextzeilen erfolgte auf Basis des natürlich und menschlich lesbaren Quelltext nach gängigen Quelltext Standards.⁸⁰

Anwendungsbereich	Menge	Programmier- oder Auszeichnungssprache
Datenbankdefinition (DDL) Datenbank ausschließlich via Oracle APEX Assistent erstellt	-	SQL
Datenbankabfrage (DML) Abfragen überwiegend durch Seitenanlage via Oracle APEX Pagebuilder initial erstellt und nur modifiziert	30 Zeilen	SQL
Oberflächengestaltung Farbanpassungen des Oracle APEX Universal Themes mit APEX Theme Roller	20 Zeilen	CSS
Autorisierungsschema APEX Prüfung ob authentifizierter Nutzer spezifischer Berechtigungsgruppe angehört	5 Zeilen	PL/SQL
Datenvalidierung und Plausibilisierung	25 Zeilen	Boolsche Operatoren, PL/SQL, SQL

⁷⁹ Best Practices basierend auf Brizzi, Ellen-Wolff & Nuijten, Oracle APEX Best Practices

⁸⁰ Vgl. Hellström, SQL Style Guide

Validierungsregeln teils initial automatisiert auf Basis der Datentypen durch Oracle APEX erstellt		
--	--	--

Tabelle 16: Individueller Quelltext in prototypischer Anwendung

Aus der geringen Menge an eingesetzten Quelltext könnte die mittelbare Schlussfolgerung gezogen werden, dass Geschäftsanwendungen ohne Quelltext in der Realisierung eine geringere Komplexität und kein IT-fachliches Wissen erfordern. Dies würde sich mit der in Kapitel 3.4 beschriebenen Anwenderzielgruppe von Low-Code Entwicklungsplattformen und dem Anspruch der Entwicklung von Anwendungen ohne Programmierer decken.

Auf Basis der Erfahrung aus Entwicklung der prototypischen Geschäftsanwendung im Rahmen dieser Thesis muss dies jedoch dahingehend relativiert werden, dass durchaus IT-fachliches Wissen erforderlich ist, auch wenn eine vermutlich vergleichsweise geringe Menge an Quelltext erforderlich ist.

Insbesondere für folgende Aspekte war IT-Fachwissen erforderlich:

- Relationale Datenmodellierung: Das Fundament der Anwendung stellt ein relationales Datenmodell dar. Auch wenn dessen Entwicklung durch Werkzeuge der Datenmodellierung und hier einem Assistenten unterstützt wurde, ist fundiertes Wissen und Verständnis von relationalen Datenstrukturen und der Modellierung in Normalformen erforderlich.⁸¹
- Verständnis von Datentypen: Auf unterschiedlichen Ebenen der Anwendung kommen Datentypen zum Einsatz (z.B. Varchar, Number, Date, ...). Die Definition der geeigneten Datentypen muss durch den Anwender der Low-Code Entwicklungsplattform erfolgen. Hier muss der Typ, Länge aber auch der Umgang mit Null-Werten und Validierungen sinnvoll gewählt werden.
- Anwendungssicherheit: Durch Konventionen und Prüfungen (z.B. Oracle APEX Security Advisor) wird der Anwender unterstützt, die Sicherheit der Anwendung sicherzustellen.⁸² Dennoch ist hierbei ein Verständnis der Authentifizierungs- und Autorisierungsmethoden und deren Einsatz im Zusammenspiel der unterschiedlichen Anwendungsschichten erforderlich. So könnte ohne Verständnis etwa der Eindruck entstehen, dass durch das reine Entfernen einer Verlinkung

⁸¹ Vgl. Martin, Application development without programmers, S. 265–266

⁸² Vgl. Spendolini, Expert Oracle Application Express Security, S. 1–3

ein unberechtigter Nutzer nicht auf eine Seite gelangen kann, was aber etwa durch Eingabe in der Adresszeile immer noch möglich ist.

Auf Basis dieser Aspekte kann die Schlussfolgerung gezogen werden, dass wenn- gleich keine explizite Kenntnis in einer Programmiersprache erforderlich ist, fundier- tes IT-Wissen in unterschiedlichen Bereichen erforderlich ist.

Das Wissen ist hierbei aber nicht so tiefgreifend erforderlich, wie es bei der Entwick- lung ohne Low-Code Ansatz nötig ist. Oft ist in der IT Praxis die Differenzierung von Programmieraufgaben auf unterschiedliche Entwicklertypen erforderlich, etwa die Aufteilung nach Front-End Entwickler und Back-End Entwickler. Bei diesen Entwick- lern handelt es sich jeweils um Spezialisten der Oberflächen- bzw. Anwendungs- und Datenbankschicht. Diese Aufteilung ist erforderlich, da die Schichten in unterschied- lichen Programmier- bzw. Auszeichnungssprachen entwickelt werden. Da durch die Low-Code Entwicklungsplattform keine tiefgreifenden Kenntnisse aufgrund der dekla- rativen bzw. visuellen Entwicklung erforderlich sind, ist es vorstellbar, dass die Diffe- renzierung dieser unterschiedlichen Fachgebiete der Anwendungsentwicklung durch den Low-Code Ansatz gemindert oder gar nicht mehr erforderlich sind.

7 Fazit

Die Auswertung des Prototyps zeigt auf, dass auf Basis der angewandten Low-Code Entwicklungsplattform eine funktionelle Geschäftsanwendung realisiert werden konnte. Gängige Anforderungen zur Erstellung einer Geschäftsanwendung wurden durch die Low-Code Entwicklungsplattform weitestgehend erfüllt oder konnten durch Umgebungslösungen realisiert werden. Zur Entwicklung war nur eine geringe Menge von Quelltext erforderlich.

Wenngleich ein direkter Vergleich von Ressourcenaufwänden, etwa die erforderliche Implementierungszeit, zwischen einer klassischen Anwendungsentwicklung und des Einsatzes einer Low-Code Entwicklungsplattform nicht Bestandteil dieser Ausarbeitung ist, besteht durchaus der abschließende Eindruck, dass die Entwicklung der prototypisch vorliegenden Geschäftsanwendung nur durch Einsatzes der Low-Code Entwicklungsplattform Oracle APEX diesen fortgeschrittenen funktionellen Zustand in kurzer Zeit erreichen konnte. Insbesondere im Bereich der Querschnitts- und Unterstützungsfunktionen konnten mit wenig Aufwand Funktionalitäten der Plattform im Entwicklungsprozess genutzt und für die Geschäftsanwendung eingesetzt werden.

Im Rahmen dieser Thesis wurden jedoch auch Nachteile bei Einsatz einer Low-Code Entwicklungsplattform identifiziert, die dazu führen, dass deren Nutzung nur wohlüberlegt und evaluiert empfohlen werden kann. Insbesondere durch den in sich geschlossenen Plattformansatz, ist ein Wechsel zwischen Low-Code Entwicklungsplattformen meist nicht möglich. Es ergibt sich daraus für Geschäftsanwendungen eine starke Abhängigkeit zu einem Anbieter, es entsteht das Risiko eines Vendor Lock-In Effekts. Somit sollte das Risiko einer Insolvenz oder Aufgabe der Low-Code Entwicklungsplattform durch den externen Anbieter mitberücksichtigt werden. Sollte durch den Anbieter die Plattform nicht weiterentwickelt oder bereitgestellt werden, stellt dies ein erhebliches Risiko für die Funktionalität der darauf basierenden Geschäftsanwendungen dar. Eine Lösung könnte hier der gezielte Einsatz von Low-Code Plattformen nur in weniger geschäftskritischen Anwendungen sein. Nicht zuletzt durch die starke und langfristige Abhängigkeit von einem Anbieter gilt es auch das Kosten- und Lizenzierungsmodell der Plattform intensiv zu prüfen. Hier kann es sinnvoll sein durch langfristige Vertragsvereinbarungen die finanzielle Nachhaltigkeit der Entwicklung sicherzustellen und gerade in lizenzbasierten Modellen plötzliche Kostenschwankungen auszuschließen.

8 Literaturverzeichnis

- Augsten, Stephan: "Was ist Low-Code Development?" <<https://www.dev-insider.de/was-ist-low-code-development-a-794952/>>(18.06.2019).
- Brizzi, Learco; Ellen-Wolff, Ilohn; Nuijten, Alex: Oracle APEX Best Practices, Birmingham: Packt Publishing, 2012.
- Bruijn, Niels de: "Affordable Workflow Options for your APEX App" <https://www.nloug.nl/downloads/ogh20170330_niels_de_bruijn.pdf>(30.05.2019).
- Cerf, Vinton G.; Cain, Edward: The DoD Internet Architecture Model. Pennsylvania State University, 1983.
- Chandrakant, Kumar: "Introduction to Flowable" <<https://www.baeldung.com/flowable>>(18.06.2019).
- Coutinho, Rodrigo S.: Bringing visual languages to market: The OutSystems story.
- Dewanto, Lofi; Klein, Manuel: "LowCode == LowQuality?" <<https://www.heise.de/developer/artikel/Low-Code-Low-Quality-4134288.html>>(14.03.2019).
- Dijkstra, Edsger W. (1972): The Humble Programmer. *Communications of the ACM*, 15 (10), 859–866.
- "Gartner IT Glossary - Citizen Developer" <<https://www.gartner.com/it-glossary/citizen-developer/>>(03.05.2019).
- Grigonis, Richard: "Generation Computers" <https://www.atariarchives.org/deli/fifth_generation.php>(30.05.2019).
- Hellström, Ian: "SQL Style Guide - Oracle SQL and PL/SQL Optimization for Developers", Oracle Corporation <<https://oracle.readthedocs.io/en/latest/sql/basics/style-guide.html>>(13.06.2019).
- Hendriks, Dré: The selection process of model based platforms. Thesis. Nijmegen: Radboud Universiteit, 2017.
- International Organization for Standardization: "Object Management Group Unified Modeling Language" <<https://www.iso.org/obp/ui/#iso:std:iso-iec:19505:-1:ed-1:v1:en>>(21.05.2019).
- Jacobs, James: "Modern Enterprise UI Design" <<https://medium.com/pulsar/modern-enterprise-ui-design-part-1-tables-ad8ee1b9feb>>(20.05.2019).
- Jahn, René: "Low-Code Development Plattform" <<https://www.informatik-aktuell.de/entwicklung/programmiersprachen/low-code-development-plattform.html>>(12.03.2019).

- Jennings, Terri: Oracle Application Express - App Builder User's Guide (19.1. Aufl.). Oracle Corporation. (28.05.2019)<<https://docs.oracle.com/en/database/oracle/application-express/19.1/htmldb/oracle-application-express-app-builder-users-guide.pdf>>, 2019.
- Kecher, Christoph; Salvanos, Alexander; Hoffmann-Elbern, Ralf: UML 2.5 - Das umfassende Handbuch, 6. Aufl., Bonn: Rheinwerk, 2018.
- Kemp, Jeffrey: "Non-Overlapping Dates Constraint"<<https://jeffkemponoracle.com/2012/08/non-overlapping-dates-constraint/>>(08.07.2019).
- Lo Giudice, Diego; Condo, Christopher: "Faster Software Delivery Will Accelerate Digital Transformation - Strategic Plan: The Modern Application Delivery Playbook"<<https://reprints.forrester.com/#/assets/2/225/RES116443/reports>>(12.03.2019).
- Lobo, Sascha: "Digitale Ungeduld"<<http://www.spiegel.de/netzwelt/web/s-p-o-n-die-mensch-maschine-digitale-ungeduld-a-774110.html>>(02.04.2019).
- Loetz, Christophe: "OSBP – die erste Software-Fabrik bei Eclipse - Ab geht's mit der Automatisierungsfabrik"<<https://jaxenter.de/osbp-die-erste-software-fabrik-bei-eclipse-77938>>(06.06.2019).
- Martin, James: Application development without programmers, Englewood Cliffs, N.J.: Prentice-Hall, 1982.
- Möckel, Karsten: "Oracle Application Express unter der Security-Lupe - APEX-Anwendungen gezielt absichern"<<https://apex.oracle.com/pls/apex/germancommunities/apexcommunity/tipp/2901/index.html>>(08.07.2019).
- Mozilla Foundation: "AJAX - Mozilla Developer Center"<<https://developer.mozilla.org/de/docs/Web/Guide/AJAX>>(28.05.2019).
- Neumann, Alexander: "BPMN 2.0 für eine bessere Zusammenarbeit zwischen Fachabteilung und IT"<<https://www.heise.de/developer/meldung/BPMN-2-0-fuer-eine-bessere-Zusammenarbeit-zwischen-Fachabteilung-und-IT-1175099.html>>(18.06.2019).
- Open Web Application Security Project: "OWASP Top 10 Risiken für die Anwendungssicherheit"<https://www.owasp.org/index.php/Germany/Projekte/Top_10_fuer_Entwickler-2013/Top_10>(18.06.2019).
- Oracle Corporation: "Licensing Information User Manual"<<https://docs.oracle.com/en/database/oracle/oracle-database/18/xelic/licensing-information.html>>(08.07.2019).
- Oracle Corporation: "Oracle APEX Statement of Direction"<<https://apex.oracle.com/en/learn/resources/sod/>>(08.07.2019).

Oracle Corporation: "Oracle Application Express Website" <<https://apex.oracle.com/de/>>(08.07.2019).

Peake, David: "Life Cycle Management with Oracle Application Express" <<https://www.oracle.com/technetwork/developer-tools/apex/learn-more/apex-life-cycle-management-wp-3030229.pdf>>(13.06.2019).

Rentrop, Christian; Augsten, Stephan. (2018): Low-Code-Entwicklung - Ansatz und Tools(30.05.2019)<<https://www.dev-insider.de/low-code-entwicklung-ansatz-und-tools-a-763889/>>.

Richardson, Clay; Rymer, John R.: "Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms - The Landscape Reflects A Market In Its Formative Years" <https://informationsecurity.report/Resources/Whitepapers/0eb07c59-b01c-4399-9022-dfc297487060_Forrester%20Vendor%20Landscape%20The%20Fractured,%20Fertile%20Terrain.pdf>(13.06.2019).

Rohleder, Bernhard: "Der Arbeitsmarkt für IT-Fachkräfte" <https://www.bitkom.org/sites/default/files/2018-12/181213_Bitkom_Charts_PK_IT-Fachkr%C3%A4fte_final.pdf>(02.04.2019).

Rymer, John R.; Koplowitz, Rob: Low-Code Development Platforms For AD&D Professionals - The 13 Providers That Matter Most And How They Stack Up. The Forrester Wave. Forrester, 2019.

Sieben, Jürgen: Oracle APEX - Das umfassende Handbuch, Bonn: Rheinwerk, 2017.

Sieben, Jürgen: Oracle PL/SQL - Das umfassende Handbuch, Bd. 6073, 3. Aufl., Bonn: Rheinwerk, 2018.

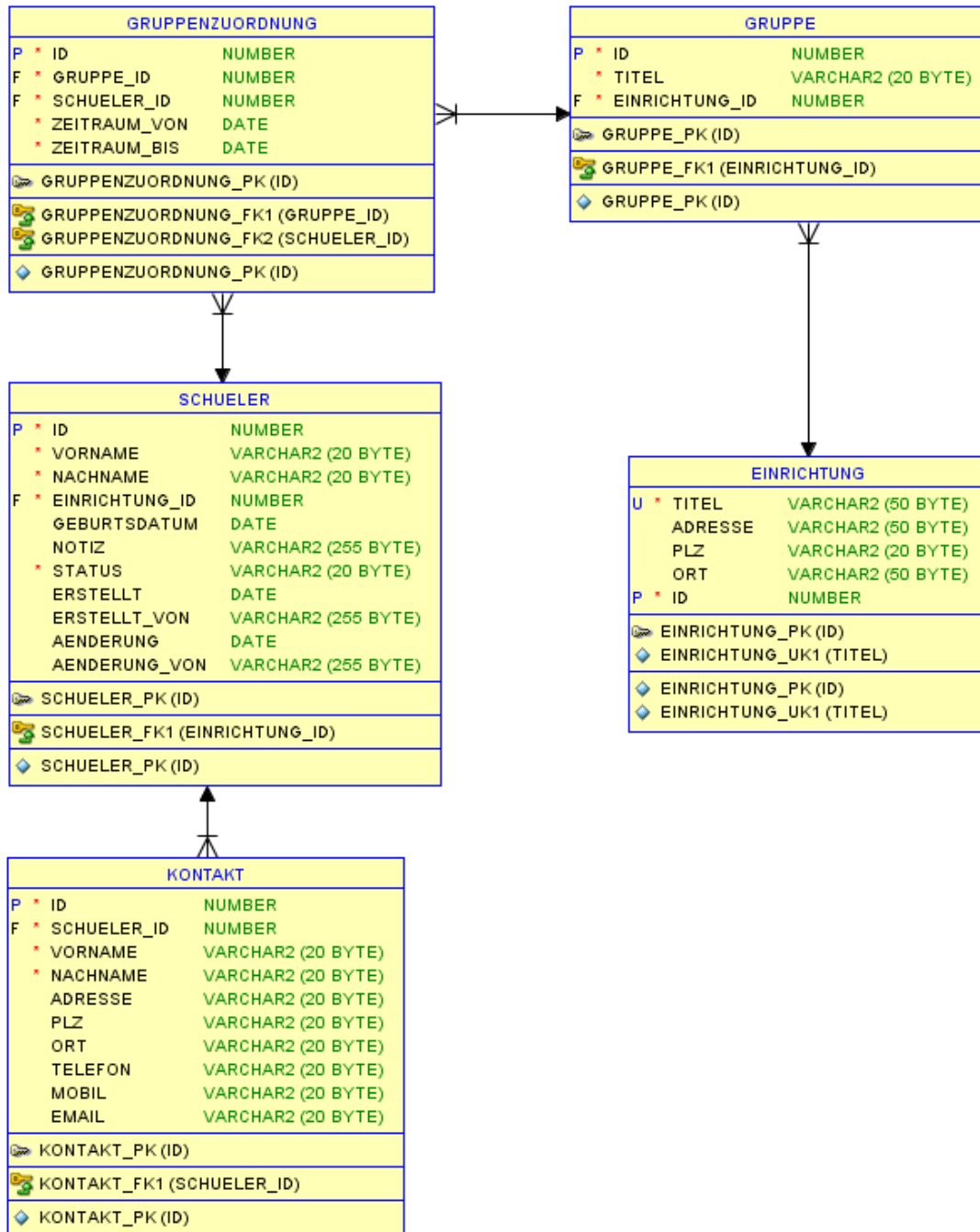
Spendolini, Scott: Expert Oracle Application Express Security, Books for professionals by professionals, New York: Apress, 2013.

Whittle, Jon; Hutchinson, John; Rouncefield, Mark. (2014): The State of Practice in Model-Driven Engineering. *IEEE Software*, 31 (3), 79–85.
<https://doi.org/10.1109/MS.2013.65>

World Wide Web Consortium: "Web Design and Application Standards" <<https://www.w3.org/standards/webdesign/>>(28.05.2019).

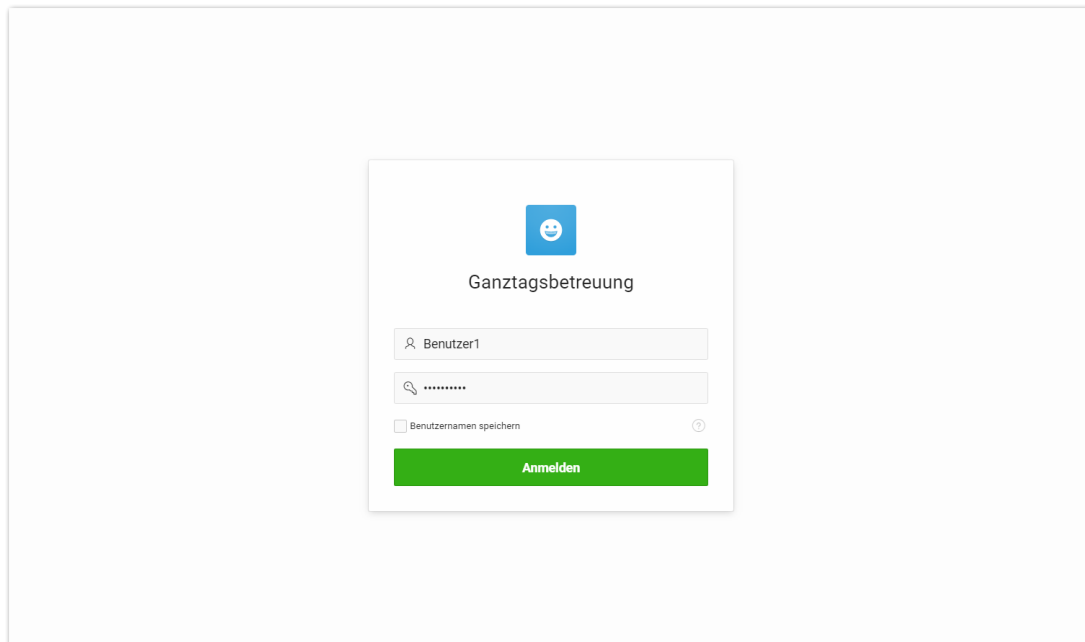
9 Anhang

9.1 Relationales Datenmodell der prototypischen Anwendung

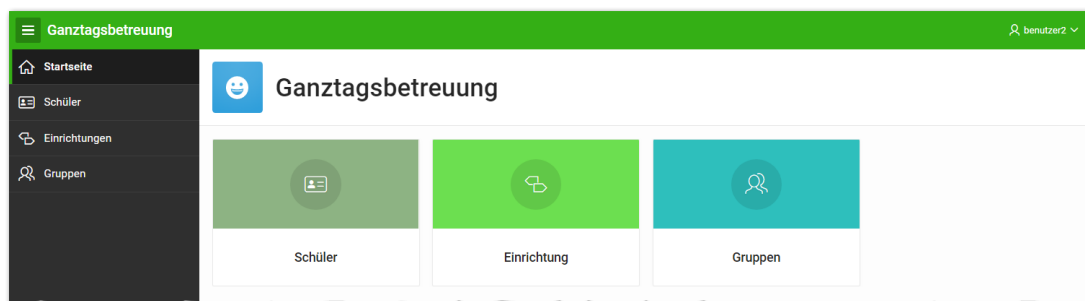


9.2 Bildschirmkopien der prototypischen Anwendung

Benutzerauthentifizierung



Startseite



Schüler Bearbeitung

Ganztagsbetreuung

benutzer2

Startseite

Schüler

Einrichtungen

Gruppen

Schüler

Suche

Huber
Johanna

Maier
Stefanie

Meier
Jonas

Meister
Max

Müller
Johannes

Muster
Franziska

Vorname
Stefanie

Nachname
Maier

Einrichtung
Mustereinrichtung

Geburtsdatum
12.01.90

Status
Bewerbung

Notiz

Abbrechen

Löschen

Änderungen anwenden

Zurücksetzen

+ Erstellen

Bearbeiten

Schüler Kontakt Bearbeitung

Ganztagsbetreuung benutzer2

Schüler Zurücksetzen + Erstellen

Suche...

Kontakt

Vorname: Hans Nachname: Mustervater

Adresse: Musterstr. 12 Plz: 50825 Ort: Köln

Telefon: +49 123 456789 Mobil: +49 161 456789

E-Mail: hans.muster@test.de

Abbrechen Löschen Änderungen anwenden

Vorname	Nachname	Adresse	Plz	Ort	Telefon	Mobil	E-Mail
Huber	Johanna						
Maier	Stefanie						
Meier	Jonas						
Meister	Max						
Müller	Johannes						
Muster	Franziska						

Schüler Gruppenzuordnung Bearbeitung

Ganztagsbetreuung benutzer2

Schüler Zurücksetzen + Erstellen

Suche...

Gruppenzuordnung

Gruppe: Mustergruppe 1

Zeitraum von: 01.01.18 Zeitraum bis: 31.12.18

Abbrechen Löschen Änderungen anwenden

Gruppe	Zeitraum von	Zeitraum bis
Mustergruppe 1	01.01.18	31.12.18

Einrichtungen Ansicht

Ganztagsbetreuung benutzer2

Einrichtungen

Suchen: Alle Textspalten Los Aktionen Bearbeiten Speichern Zeile hinzufügen Zurücksetzen

	Titel	Adresse	Plz	Ort
<input checked="" type="checkbox"/>	Mustereinrichtung	Musterstr. 12	50825	Köln
<input type="checkbox"/>	Mustereinrichtung 2	Musterstr. 14	50558	Köln

1 Zeilen ausgewählt Gesamt 2

Einrichtungen Erfassung mit Validierungsfehler

Ganztagsbetreuung benutzer2

Einrichtungen

Suchen: Alle Textspalten

<input type="checkbox"/>	<input type="checkbox"/>	Titel	Adresse	Plz	Ort
<input type="checkbox"/>	<input type="checkbox"/>	Mustereinrichtung	Musterstr. 12	50825	Köln
<input type="checkbox"/>	<input type="checkbox"/>	Mustereinrichtung 2	Musterstr. 14	50558	Köln
<input checked="" type="checkbox"/>	<input type="checkbox"/>		Musterstr. 16	50825	Köln

1 Zeilen ausgewählt Gesamt 3

Betreuungsgruppen Ansicht

Ganztagsbetreuung benutzer2

Gruppen

Suchen: Alle Textspalten

<input type="checkbox"/>	<input type="checkbox"/>	Titel	Einrichtung
<input type="checkbox"/>	<input type="checkbox"/>	Mustergruppe 1	Mustereinrichtung
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Mustergruppe 2	Mustereinrichtung
<input type="checkbox"/>	<input type="checkbox"/>	Mustergruppe 3	Mustereinrichtung 2

1 Zeilen ausgewählt Gesamt 3

Erklärung

Ich versichere die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum

Rechtsverbindliche Unterschrift